UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Simulación de escenarios de crowd-sensing para analizar la credibilidad de la información de fuentes sociales

Autor: Alberto García-Muñoz Fernández-Calvillo

Tutor: Andrés Marín López

Fecha
22 de junio de 2016

Los contenidos dispuestos a lo largo del presente documento han sido redactados conforme a la normativa establecida. No obstante, con el motivo de presentar una estructura homogénea dentro de la memoria, los capítulos *Introducción* y *Conclusiones y trabajos futuros* en su versión en inglés han sido dispuestos como capítulos del apéndice.

Agradecimientos

En primer lugar me gustaría agradecer a mi padre y a mi madre el apoyo incondicional que he tenido a lo largo de mi vida académica y que se ha visto acrecentado durante mi etapa universitaria. Ya no solo quiero agradeceros ese apoyo económico sin el cual este trabajo no sería posible, sino que lo más importante es el clima que habéis dispuesto para que pueda estudiar y trabajar cómodo.

Por otro lado me gustaría agradecerles a mi hermana y a mi hermano la capacidad que hemos tenido para soportarnos y entendernos a lo largo de tantos años. Los tres hemos sido estudiantes y sabemos que durante el año tenemos periodos en los que estamos mejor y peor de ánimo. Pese a esto hemos sabido apoyarnos entre nosotros.

También quiero agradecer a el tutor de este proyecto, Andrés Marín, por haberme ofrecido la oportunidad de realizar este trabajo allá por Octubre bajo su tutela destacando la paciencia y los consejos recibidos sin los cuales no hubiese sido posible escribir esto hoy. No quiero olvidar tampoco a Florina Almenares quien ha hecho este proyecto posible.

A mis amigos Jorge, David, Fernando, Víctor, María y compañía con los cuales he pasado momentos inolvidables, aunque en el caso de que algunos nos veamos cada mucho tiempo, siempre es bueno despejarse y olvidarse de los problemas que tiene uno con vosotros. En especial quiero agradecerle esta tarea a Manuel quien ha sido el principal apoyo que he encontrado dentro de la carrera y es de los pocos que seguimos al pie del cañón trabajando.

A toda esa gente que he conocido a lo largo de estos años y que han llegado para quedarse y a los que tristemente se fueron, también mil gracias.

Si me olvido de alguien espero que me perdone, pero quiero que seáis conscientes de que sin vosotros nada de esto hubiese sido posible.

¡Muchas gracias a todos!.

Resumen

Las redes sociales son una de las mayores fuentes abiertas de información que existen actualmente en Internet. De entre todas destaca Twitter. Twitter es la cuarta red social en número de usuarios activos y continúa creciendo. Sus características la hacen ideal para la retransmisión y propagación de información de forma inmediata.

Los usuarios son los que aportan con sus tweets (mensajes que se publican en Twitter de 140 caracteres) a la creación de una enorme cantidad de datos, siendo obligatorio la búsqueda y categorización por medio de procesos automáticos.

Sin embargo, es el contenido de los tweets el que plantea mayores retos a los investigadores, pues su clasificación resulta bastante difícil. Ante el valor que supone conocer la opinión de la sociedad, universidades y empresas están dedicando gran cantidad de recursos al estudio y desarrollo de nuevos métodos de análisis automático de la información para estas plataformas

Por otro lado, los contenidos de los tweets no siempre comunican la verdad, por lo que es necesario aportar una herramienta capaz de medir la credibilidad del contenido. Es aquí donde se enmarca el proyecto, proponiendo el programa *Sniffer* para ayudar a realizar esta labor. Este programa será capaz de proponer una clasificación de la credibilidad de los distintos tweets dentro del contexto en el que se encuentren.

Palabras Clave: Social sensing, Redes Sociales, Twitter, Credibilidad, Categorización

Abstract

Social networks are one of the major open sources of information that currently exist on Internet. Amongst them, Twitter stands out. Twitter is the fourth social network by number of active users and it is still growing. Its characteristics make it ideal for retransmission and propagation of information in real time.

Each user contributes to create this huge amount of data with its tweets (messages of up to 140 characters that are posted on Twitter) making it necessary to develop automated search and categorization algorithms.

However, the information that these messages are transmitting poses a great challenge for analysts because sometimes their categorizations are so difficult. Given the value of such opinions representing the views of the society, universities and companies are devoting significant resources to research and develop new methods of information analysis for these platforms (Social Networks).

It is essential to emphasize that the content of these tweets do not always communicates real messages, making it necessary to develop some tools capable of measuring the credibility of the content of the tweet. The *Sniffer* project proposes an automatic way for helping to make this task easier. This program will be able to help on the classification of the credibility of every tweet according to the context that it has been posted.

Keywords: Social sensing, Social Networks, Twitter, Credibility, Categorization

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Este capítulo proporciona una introducción al contexto en el que este proyecto se enmarca, describiendo las razones que han motivado su realización y los objetivos que se plantea alcanzar. Por último, se presenta un resumen de los contenidos de cada capítulo y apéndice que constituye este documento.

1.1 Motivación del trabajo

A lo largo de la historia, la información ha procedido a través de los denominados *servicios tradicionales*, estos servicios han sido la televisión, la radio o la prensa escrita. Sin embargo a partir de la revolución digital, y más concretamente con la aparición de los smartphones o teléfonos inteligentes la sociedad ha comenzado a realizar uso de las denominadas redes sociales (RRSS) como su principal fuente de información.

Para comenzar, una red social consiste en una estructura integrada por personas, organizaciones o entidades conectadas entre sí por medio de relaciones como: relaciones de amistad, parentesco, económicas o intereses comunes. Este término se ha actualizado en los últimos años señalando a un tipo de sitio de Internet que favorece la creación de comunidades virtuales, en las cuales es posible acceder a servicios que permiten crear grupos sociales según los intereses del usuario, compartiendo fotografías, videos e información en general. Diferentes ejemplo de redes sociales son Facebook[1], Twitter[2] o YouTube[3].

[1]https://www.facebook.com
[2]https://twitter.com
[3]https://www.youtube.com/

A nivel global, existen infinidad de redes sociales, sin embargo, hay dos que muestran un claro dominio sobre el resto. Twitter y Facebook representan los máximos exponentes del cambio hacia los nuevos modelos de comunicación. El caso de Facebook agrupa a más de 1.300 millones de usuarios activos alrededor del mundo [1], sin embargo, esta plataforma es menos atractiva, pues la información que se transmite más asiduamente a través de la misma es de carácter personal y no de carácter informativo. Por otro lado se presenta la red social Twitter, este entorno está formado por más de 284 millones de usuarios activos y alcanza en total más de 600 millones, pese a tener un número de usuarios menor que Facebook es interesante pues es la principal fuente de noticias, los grandes informadores a nivel global tienen presencia muy activa en esta red además de multitud de organismos nacionales.

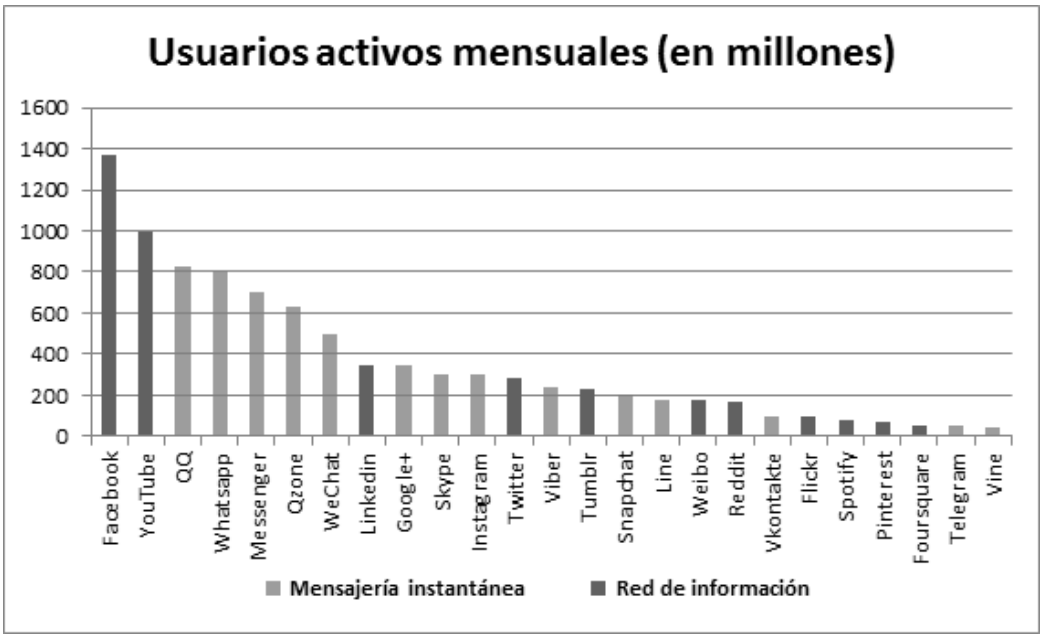

Figura 1.1: Distribución del número de usuarios activos en las Redes Sociales Fuente: *http://www.marketingandweb.es/*

Conforme este tipo de plataformas han obtenido usuarios, han adquirido una posición privilegiada tanto en la creación como en la difusión de noticias debido a que cada usuario se comporta como una fuente de noticias y la malla que establece en torno a él, ya sean amigos (Facebook) o seguidores (Twitter); componen los elementos de distribución de la información. Con esto se permite que una información se propague de forma casi instantánea.

Es este el principal motivo por el que en momentos de crisis, desastres naturales o llamamientos de urgencia; estos canales se están convirtiendo en elementos cruciales para la difusión de información útil a los usuarios afectados.

No obstante, debido a la enorme facilidad de propagación de información en redes sociales aparece un problema crucial. Este problema se presenta ante la enorme cantidad de información a la que se enfrenta el usuario. Esta información ha sido emitido sin ningún control por parte de los usuarios ni de los administradores, a diferencia de los medios tradicionales cuya información está revisada y contrastada con la finalidad de que sea lo más verídica posible. Por tanto, cuando el usuario se dispone a obtener la información en este tipo de plataformas, necesita que previamente se realice un análisis de la información. Este análisis está ayudando al usuario a discernir en torno a la veracidad del contenido, facilitando la detección de bulos o informaciones falsas que se propagan sin control a lo largo de la red.

Un ejemplo claro de lo que se puede considerar como información creíble e información poco creíble son las capturas procedentes de la red social Twitter representadas en la figura 1.2. En estas capturas se muestran dos mensajes (Tweets), de los cuales el primero se puede considerar como creíble, por distintos motivos: el usuario que lo publica es un organismo oficial, no utilizan palabras extrañas, está escrito con todos los elementos lingüísticos (tildes, comas) y aparte tiene una gran repercusión en la red social: más de 7.700 redifusiones (retweet) y más de 2.500 me gusta (Favs). Por el contrario, el segundo tweet no genera la misma confianza que el anterior, ya sea porque no se posee información acerca de la autora, por la presencia de diferentes elementos extraños como la segunda etiqueta (Hashtag) o su poca relevancia dentro del contexto, el tweet no tiene redifusiones ni me gustas.

Figura 1.2: Ejemplo de tweets

Esta tarea, que parece muy sencilla de realizar para las personas, presenta una dificultad muy elevada para ser realizada con máquinas debido a que la información que se presenta no posee una estructura estable o prefijada. Por tanto esta tarea es la que se pretende desarrollar durante este documento, la proposición de un sistema que permita ayudar a la extracción de la credibilidad de los mensajes publicados en Twitter. La principal razón por la que se necesita implementar esta tarea de forma automática, es que a diario se publican más de 320 millones de tweets de diversa naturaleza en Twitter. Por consiguiente, el usuario de esta red social no puede estar comprobando constantemente cuales son los tweets verídicos o creíbles, humanamente esta tarea es imposible. Para que la estancia del usuario en la red social sea lo más cómoda posible, es necesaria la implantación de algún tipo de herramienta la cual le realice dicha clasificación. Con estas herramientas, el usuario es capaz de comprobar directamente si el contenido del mensaje es creíble o no es creíble dentro de su contexto.

Cabe remarcar esto último que se ha mencionado. No se puede determinar si un contenido es creíble o no lo es de forma aislada. Si se considera cada Tweet de cada contexto de forma aislada se van a eliminar todas las relaciones existentes con el resto de mensajes, por tanto se va a perder la información de ese mensaje con el contexto en el que se ha publicado. No es lo mismo que se considere a un mensaje aislado como creíble cuando un usuario afirma la información *"Hay un terremoto en Madrid"* y que cuando se considere en su entorno la mayoría de usuarios emita *"No hay un terremoto en Madrid"* y el contexto haya comenzado hace 3 días. Es decir, un mensaje no dice lo mismo de forma aislada que cuando se interpreta dentro de su contexto con otros mensajes.

Como se ha intentado mostrar a lo largo de estas líneas, el determinar la credibilidad de la información que se está recibiendo, que es para lo que está destinado este trabajo; puede ser útil para:

- La recolección o el modelado de los datos de proyectos en torno a análisis de contenidos, pues es de gran ayuda el poder establecer un primer corte entre los tweets que son creíbles o no creíbles, a fin de poder extraer un conjunto de muestras óptimo en el contexto en el cual se quiera trabajar.

- El hecho de que el usuario pueda ser una entidad gubernamental, permite que pueda *"extraer"* información realmente valiosa de las redes sociales, por ejemplo, acerca de ciertos incidentes que puedan afectar

la vida de una sociedad, para ser compartidos con un cierto grado de fiabilidad.

- En situaciones de emergencia social, para ayudar a los servicios de urgencia y afectados en tareas de coordinación.
- Conocer la percepción más verídica entre los usuarios de la red social sobre las actuaciones de una determinada entidad, desde personas físicas a empresas.

1.2 Objetivos

Con motivo de lo explicado anteriormente, el principal objetivo de este proyecto va a ser a creación de un sistema que sea capaz de ayudar en la clasificación automática de la credibilidad de los mensajes (tweets) emitidos por todo tipo de usuarios en la red social Twitter. Estos mensajes no se van a analizar de forma aislada, el análisis se realizará en base al contexto en el que se están emitiendo. A este contexto será a lo que se referirá a lo largo del proyecto como tendencia (o Hashtag en la jerga de Twitter).

A lo largo de las siguientes páginas, se expondrán una serie de etapas por las que se ha ido creando el proyecto. Cada una de estas fases tiene su propio objetivo, sin embargo; sumando todos estos objetivos parciales se obtiene el objetivo final del proyecto que se acaba de exponer. Los objetivos parciales para las diferentes etapas son:

- La realización de un análisis de las variables que presentan una mayor influencia en la clasificación del tweet. (Capítulo 4)
- La realización de la extracción y almacenamiento de los tweets en base al contexto en el que están siendo emitidos. (Capítulo 5)
- La realización de la clasificación de los tweets en función de las variables mencionadas anteriormente.(Capítulo 5)
- La realización de un sistema para la representación de los resultados con la finalidad de comprender como es el contexto en el que se emite el tweet. (Capítulo 6)

Este sistema por tanto proporciona una operatividad completa, desde la recolección de los mensajes o tweets hasta la clasificación y la representación de los resultados. Más adelante, estos objetivos se expondrán con más detalle y se desarrollará su funcionamiento.

La pregunta que se pretende ayudar a responder a lo largo del presente documento es: **¿Cómo se puede extraer la credibilidad en Twitter?**. Como parte de la solución, se plantea el sistema *Sniffer*. Este sistema es una contribución al problema de la credibilidad expuesto a lo largo de este capítulo.

1.3 Contenido de la memoria

Este documento constituye la memoria del proyecto. Esta memoria está estructurada en los siguientes capítulos:

- **Capítulo 2**: A lo largo de este capítulo se va a tratar de exponer la situación actual del campo en el que se va a desarrollar el proyecto. Se expondrá la evolución del campo a lo largo de las distintas tendencias que existen y por último se elaborará una breve conclusión con la que se expondrá la situación inicial del proyecto.

- **Capítulo 3**: A lo largo de este capítulo se trata de exponer los límites de uso que se puede hacer con los datos procedentes de Twitter, enmarcándolos en la política de Twitter y a nivel europeo. El motivo de realizar este punto separado es que se van a utilizar datos de índole personal.

- **Capítulo 4**: En este capítulo se trata de exponer la base teórica entre las relaciones de las diferentes variables de los metadatos del tweet para la posterior implementación en el programa *Sniffer*.

- **Capítulo 5**: En este capítulo se trata de exponer la estructura y algoritmos para la fase de clasificación del programa *Sniffer*.

- **Capítulo 6**: En este capítulo se trata de exponer la estructura y algoritmos para la fase de visualización del programa *Sniffer*.

- **Capítulo 7**: En este capítulo se trata de exponer las diferentes pruebas realizadas al programa *Sniffer*.

- **Capítulo 8**: En este capítulo se trata de exponer las diferentes etapas por las que ha transcurrido el programa *Sniffer*.

- **Capítulo 9**: En este capítulo se trata de exponer las principales conclusiones y posibles trabajos futuros asociados al programa *Sniffer*.

- **Apéndice B**: En este capítulo se va a desarrollar un resumen del proyecto, además se van a exponer los principals hitos que presenta.

- **Apéndice D**: En este capítulo se va a tratar de explicar cada una de las tecnologías que se han utilizado en el desarrollo del programa *Sniffer*.

- **Apéndice E**: En este capítulo se va a exponer el proceso de instalación y despliegue del programa *Sniffer*.

- **Apéndice F**: En este capítulo se va a exponer el manejo del programa *Sniffer*.

Capítulo 2

Estado del arte

A lo largo de este capítulo se va a tratar de exponer la situación actual del campo en el que se va a desarrollar el proyecto. Se expondrá la evolución del campo a lo largo de las distintas tendencias que existen y por último se elaborará una breve conclusión con la que se expondrá la situación inicial del proyecto.

2.1 Introducción al crowd-sensing y social sensing

Se conoce como *crowd-sensing* a la disciplina que se basa en la utilización de multitud de dispositivos y sensores para la recolección de los datos [2]. Esta disciplina es utilizada para adquirir conocimientos a través de sensores integrados en una serie de dispositivos, los cuales son utilizados por las personas, para extraer información con la finalidad de medir y caracterizar fenómenos de interés común. Esta disciplina es una de las bases del denominado *"Internet de las cosas"* (IoT)[3], pues uno de los objetivos que pretende cumplir esta tecnología es disponer una serie de sensores para poder caracterizar diferentes fenómenos que ocurran en el entorno de las personas.

La realización y el éxito de un proyecto de *crowd-sensing* se basa en 4 etapas principales según el autor de [4]:

1. Establecimiento de las metas: Esta fase implica la definición del modelo estudiado, la definición de los criterios esenciales y las unidades estadísticas examinadas. También es necesario establecer las dimensiones del proyecto y una planificación del mismo. En esta fase también

se debe proporcionar la utilización operativa de la información y la extracción de los modelos de productos así como la especificación de los resultados esperados.

2. Elección de los sensores: Esta fase consiste en la elección de herramientas fiables, de buena reputación y adecuados

3. Recopilación de la información: esta fase tiene como objetivo extraer la información de los fenómenos determinados que se deseen analizar en el proyecto

4. Utilización de la información: Esta fase consiste en la aplicación de diferentes técnicas tales como: segmentación, generación de nuevos datos... Los cuales permitan extraer el conocimiento necesario para cumplir los objetivos establecidos.

Actualmente existe una tendencia que está tomando cada vez más fuerza dentro de esta disciplina, esta tendencia se caracteriza por considerar a las personas como las principales fuentes de información. Esta tendencia es la que se denomina como *social sensing*[5], es decir, valerse de fuentes sociales como base para la obtención de información.

2.1.1 La credibilidad en entornos de social sensing

Esta disciplina presenta un problema importante dentro del proceso de recolección de los datos: no todas las personas poseen los mismos umbrales de decisión ante las mismas acciones. Por tanto, la percepción de las acciones, que es al final el dato que se recoge; puede estar tergiversado por la fuente. Este es el motivo por el que se entiende que las personas se presentan como sensores con una menor fiabilidad que los que han sido diseñados para un propósito específico [5].

Es por ello por lo que la información que procede de estas fuentes debe de ser modelada para su correcta manipulación. Charu C. Aggarwal, autor de *"On credibility estimation tradeoffs in assured social sensing"*[5], considera que se debe de establecer una primera etapa en el modelado de los datos. Como base para esta primera etapa, se define la utilización del término "credibilidad". Este autor define el término "credibilidad"de la siguiente manera: *"Ofrecer motivos suficientes para ser creído"*.

2.1.2 Entornos de social sensing

El *social sensing* es una disciplina puede ser entendida en multitud de entornos, a lo largo del artículo [5] se expone como puede ser útil para la detección de semáforos en rojo y así evitar atascos. Otra de las aplicaciones sirve para la localización de las personas por medio del dispositivo GPS que incorporan en sus teléfonos móviles, como expone el autor de [2].

Sin embargo, en la actualidad están proliferando en Internet una serie de plataformas conocidas como redes sociales. Estas redes sociales se fundamentan en el intercambio de información, y por consiguiente, se convierten en uno de las principales entornos de *social sensing*, donde cada uno de los usuarios se va a comportar como una fuente de información[6].

Existen una enorme cantidad de redes sociales. Acorde con la clasificación realizada por el autor de [7], estas redes se pueden clasificar en 2 tipos en función del contenido que se comparta en ellas:

- **Generalistas**: la finalidad de este tipo de redes sociales es comunicar a sus usuarios permitiéndoles transmitirse información de cualquier propósito y sobre cualquier formato. Entre estas redes están Facebook[1], Twitter[2] o Google+[3].

- **De propósito específico**: los usuarios de este tipo de redes comparten información en un formato específico, ya sea vídeo como YouTube[4] o imágenes como Instagram[5].

Por otro lado aparecen diferentes formatos en la interacción entre los usuarios. Apareciendo así dos tipos de redes:

- **Seguimiento Mutuo**: Este tipo de redes se caracterizan por que cuando un usuario "A" comienza "seguir" o ser "amigo" de otro usuario "B" el usuario "B" automáticamente se convierte en "seguidor" o "amigo" de "A". Este tipo de estructuras sitúan a los ambos usuarios al mismo nivel jerárquico dentro de la cadena de difusión de los mensajes, por lo que no suelen ser utilizadas en gran medida por entidades corporativas para difundir sus mensajes.

[1] https://www.facebook.com
[2] https://www.twitter.com
[3] https://plus.google.com
[4] https://www.youtubecom
[5] https://www.instagram.com

- **Seguimiento Independiente**: Este tipo de redes se caracteriza por lo contrario a la anterior: cuando el usuario "A" sigue a "B", "B" no está obligado a devolver el seguimiento. Es por esto por lo que este tipo de estructuras se asemejan a redes de difusión, por ello suelen ser muy utilizadas por las entidades corporativas para difundir sus mensajes.

De entre todas las redes sociales, Twitter[6] se ha convertido en una de las más importantes para esta disciplina. Esto es debido a que los mensajes (Tweets) que se publican son cortos y, por tanto, se puede transmitir una información muy concisa y rápida. Es por ello por lo que han aparecido estudios de diversa índole, desde el análisis del sentimiento de los tweets [8] hasta sistemas que pretenden analizar la credibilidad de los contenidos que se publican [9, 10].

2.2 Twitter como fuente de información

Twitter constituye una red de seguimiento independiente, es decir, el hecho de que un usuario *"A"* siga a un usuario *"B"* no implica que *"B"* deba de seguir a *"A"*. Es por este motivo por el que en esta red proliferan perfiles de usuario relacionados con empresas u organismos públicos los cuales tienen unas políticas de comunicación basadas únicamente en la difusión de información a sus seguidores [11].

Debido a la aparición de este tipo de perfiles dedicados a la difusión, se podría llegar a establecer una diferenciación dentro del conjunto de usuarios. Una primera diferenciación se puede entender como en el marco de los medios de transmisión de información tradicionales (Televisión, radio o periódicos), donde los usuarios pueden ser de dos tipos:

- Generadores de contenido: En los medios tradicionales este tipo de usuarios se han caracterizado por generar un contenido, ya sea de carácter informativo o no; para un gran número de oyentes (consumidores de contenido). En Twitter ocurre exactamente lo mismo, este tipo de usuarios se caracteriza por tener un número mayor de seguidores que de seguidos

- Consumidores de contenido: Son el tipo de usuarios hacia los cuales se dirige la información que emiten los generadores. Los usuarios pertenecientes a este conjunto son vulnerables de recibir la información creíble o la no creíble por parte de cualquier usuario.

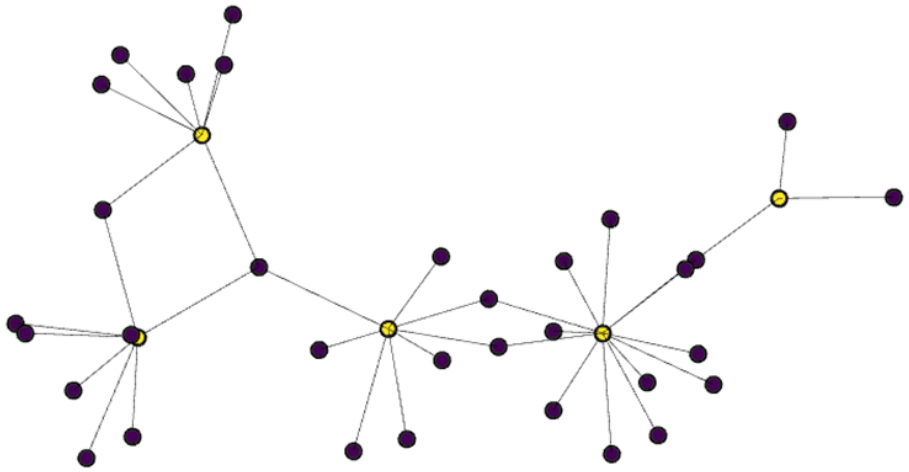

Figura 2.1: Diagrama de influencias (Amarillo consumidores de contenido, Morado generadores de contenido)

Sin embargo, se está suponiendo que la red social Twitter es un entorno donde la información circula de los generadores clásicos de contenido hacia los consumidores clásicos de contenido. Esta afirmación es absolutamente incorrecta, acorde a la exposición que se realiza a lo largo del artículo [12], el principal motivo para rebatir esta afirmación es que en Twitter cada usuario puede publicar la información de lo que está ocurriendo de su propia mano, es decir todos los usuarios se convierten en generadores de contenido; esta es la clave del éxito no solo de Twitter, sino del resto de redes sociales. Una frase que puede simplificar toda esta exposición es la siguiente:

"Todos los usuarios en Twitter se convierten en generadores y a su vez consumidores de contenido"

[6]https://twitter.com

2.3 Análisis de la información en Twitter

Los mensajes que se publican en Twitter se denominan *Tweets*, este tipo de mensajes tiene una peculiaridad: su longitud no puede exceder los 140 caracteres. Es este el principal motivo por el cual cada uno de estos mensajes se puede entender como que presenta una información limitada.

Con la finalidad de realizar un análisis en torno al contenido de los tweets, el autor de [7] plantea una de las vertientes que tiene el análisis de este tipo de textos (textos cortos), el análisis semántico del lenguaje natural.

En todos los análisis semánticos de textos es necesario extraer representaciones formales de las palabras (lemas), especialmente en el castellano. Estas representaciones suelen asociarse con el número de ocurrencias en el texto (frecuencias relativas). Es aquí donde aparecen una enorme cantidad de modelos probabilísticos y procedimientos iterativos.

Figura 2.2: Ejemplo de lematización

Sin embargo, los métodos de análisis de semántico presentan la percepción de los textos como sucesiones de palabras, pero no estudian cuál es el sentido del mismo. Por lo que se puede considerar que este tipo de métodos están muy limitados a la hora de satisfacer las necesidades de los clasificadores ante la presencia de textos cortos.

La propuesta realizada por este autor [7] indica que a la hora de realizar la clasificación de los tweets, al ser una longitud tan escasa, es prácticamente imposible realizar la extracción del sentimiento del contenido, debido a que el sistema es incapaz de detectar figuras lingüísticas tales como ironías, sarcasmos o burlas.

Es en este punto donde se puede llegar a plantear la utilización de los datos presentes en los tweets para realizar el análisis del contenido de los mismos. Estos datos son lo que se entiende por *metadatos* del tweet. Este es el punto de partida de los autores de [9, 10] quienes, como se verá a

continuación, pretenden basar la extracción de la credibilidad del contenido en función de los metadatos que son posibles extraer de los diferentes tweets que recolecten.

2.4 La credibilidad en Twitter

En la actualidad, existe una extensión denominada *TweetCred* la cual se puede añadir a Twitter y que representa el resultado del estudio [10]. Los autores de este estudio plantean una herramienta en tiempo real capaz de realizar la clasificación de los tweets que un usuario recibe en su tablón principal. El modelo que es planteado por estos autores se basa en una máquina de soporte vectorial (SVM) para realizar la clasificación.

La peculiaridad de este sistema es que, al ser una herramienta que trabaja en tiempo real, debe de primar el tiempo de respuesta para no afectar a la experiencia del usuario en la red social. Por tanto, prioriza el tiempo de respuesta del sistema ante la entrada de un nuevo tweet en detrimento de las prestaciones de la clasificación del sistema.

Por otro lado, los autores del estudio [9] plantean un modelo diferente al anterior. Estos autores establecen un árbol de decisión como la base para su clasificación, sin embargo esta elección viene determinada por que su sistema no pretende trabajar en tiempo real, sino con conjuntos de muestras extraídos y almacenados previamente.

Un punto en común que presentan ambos estudios, es que basan todo el análisis del contenido en función de los metadatos que tienen asociados los tweets. Esta información adicional asociada al contenido no solo está relacionada con el contenido que se está transmitiendo. Es por ello por lo que realizan una clasificación de la procedencia de la diferente información:

- **Información del contenido**: Este conjunto de los metadatos es lo que más está relacionado con el análisis semántico del mensaje. Sin embargo no aplica ninguna técnica relacionada con esta metodología, únicamente pretende obtener frecuencias relativas de determinadas secuencias de caracteres.
- **Información del usuario**: Este conjunto de los metadatos tiene relación con el usuario. Una parte importante de la credibilidad procede del usuario, es por ello por lo que se debe de extraer la información necesaria del usuario que pueda ser determinante en la credibilidad.

- **Información del contexto**: Este conjunto de metadatos representa al contexto en el que se desenvuelve el mensaje. El objetivo es obtener la variación que presenta el tweet a analizar con el contexto al que pertenece.
- **Información de la propagación**: Este conjunto de metadatos guarda relación con cómo se comporta el tweet dentro del contexto en el que se desarrolla su actividad. Es decir, trata de extraer la relevancia que ha adquirido el mensaje en el contexto.

Las ventajas que tiene el sistema que se plantea en [9] son las siguientes: se presentan un algoritmo de decisión bastante intuitivo basado en un árbol de decisión, los resultados de la clasificación son fáciles de comprender. Sin embargo, el modelo de clasificación que implementan es demasiado simple, debido a que el sistema que está planteado parece que no tiene una primera clasificación de los datos para extraer los que poseen información. Por tanto, los autores suponen que todos los tweets que se publican en la red social contienen información. Esto puede arrojar ciertos problemas debido a que no se debe de juzgar la credibilidad de mensajes que no contienen información.

Por otro lado, el sistema que se plantea en [10] sí que realiza la selección de los tweets que poseen información. Con esto se está evitando reducir el conjunto que se quiere analizar y por tanto pueden mejorar las prestaciones. Sin embargo, el planteamiento de realizar la plataforma para que desarrolle su actividad en tiempo real lleva a seleccionar clasificadores subóptimos, si se comparan las prestaciones obtenidas con las obtenidas por [9], con tal de ganar prestaciones en lo que a tiempo de respuesta se refiere. Además, la elección de los clasificadores que se realiza presenta una mayor dificultad de comprensión debido a que, al no presentarse una clasificación binaria como en el caso anterior, es necesario realizar una transformación a los conjuntos [13].

Se ha expuesto un término que presenta una importancia en este tema, la relevancia. Respecto a este término, se ha elaborado recientemente el estudio [14] en el cual se plantea qué es necesario para conseguir obtener el grado influencia en Twitter y cómo se desenvuelve un usuario dentro de una temática concreta.

Este estudio parece seguir la estructura clásica de difusión de la información mencionada anteriormente, sin embargo, no se está restringiendo a que el generador de la información es el que mayor proporción de seguidores-seguidos posea. Esta forma de tratar la relevancia permite

que cuentas con una relación seguidores-seguidos muy alta sea muy relevante en un cierto contexto mientras que en otro sea totalmente transparente.

La relevancia que adquiere un mensaje en el contexto nunca debe de ser el factor determinante en la asignación de credibilidad a los mensajes. Sin embargo, puede ser útil para realizar un análisis de la repercusión que han tenido los diferentes tweets clasificados.

2.5 Autenticación en Twitter

Desde 2013 se ha producido un cambio en cómo se establecen las relaciones entre las aplicaciones y los Servicios de Twitter. Hasta esa fecha, las peticiones se podían autenticar por medio del nombre de usuario y la contraseña del que fuese propietario de la aplicación o bien mediante OAuth[7]. Sin embargo, a partir de 2013 la única forma de autenticación es por medio del protocolo OAuth. Este mecanismo, representado en la figura 2.3, es un protocolo de código abierto el cual permite obtener una autorización segura para el uso de una API sin necesidad de tener que validarse con el usuario y la contraseña.

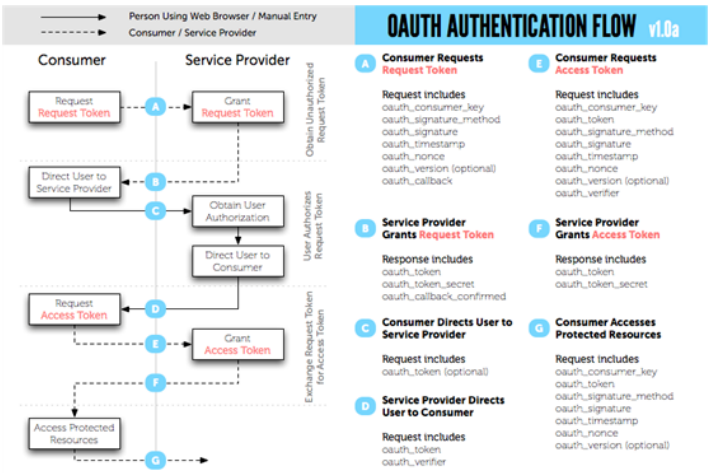

Figura 2.3: Flujo de Autenticación de OAuth Fuente: https://www.paradigmadigital.com/dev/spring-social/

[7]http://oauth.net

Las ventajas que se obtienen al implementar este sistema de validación es que los usuarios pueden revocar los accesos a la aplicación sin tener que estar vinculados con su cuenta. También permite que las aplicaciones no deban de presentar el nombre de usuario y la contraseña en la aplicación. Las diferentes API que ofrecen los Servicios de Twitter ya tienen implementado este mecanismo de validación. Para poder iniciar el proceso de autenticación, únicamente de presentar los elementos que figuran en la tabla 2.1[15].

Parámetro	**Descripción**
Consumer Key	Este elemento asemeja al antiguo nombre de usuario, es común a todas las aplicaciones de un mismo usuario y sirve para validar al usuario en los Servicios de Twitter.
Consumer Secret	Este elemento asemeja a la antigua clave de usuario, es común a todas las aplicaciones de un usuario y junto con el Consumer Key permite validar al usuario en los Servicios de Twitter.
Access Token	Este elemento es lo que se asemeja al nombre de la aplicación en los Servicios de Twitter. Este elemento se puede regenerar y aportarle o revocarle los permisos que tiene asociados.
Access Token Secret	Este elemento es lo que se asemeja a la clave de validación de la aplicación en los Servicios de Twitter. Al igual que con el Access Token, permite completar la validación de la aplicación en la plataforma.

Tabla 2.1: Parámetros de Autenticación OAuth

Estos elementos se generan automáticamente cuando el usuario, registrado como desarrollador; crea una nueva aplicación[8]. Estas claves de acceso, como ya se ha comentado, se pueden revocar en cualquier momento por parte del usuario accediendo a la App que se desee.

Hay que destacar que toda esta validación se debe de realizar de forma previa a trabajar con el resto de la API, debido a que si no se realiza no se va a tener una autorización para poder trabajar con los Servicios de Twitter.

[8]https://apps.twitter.com

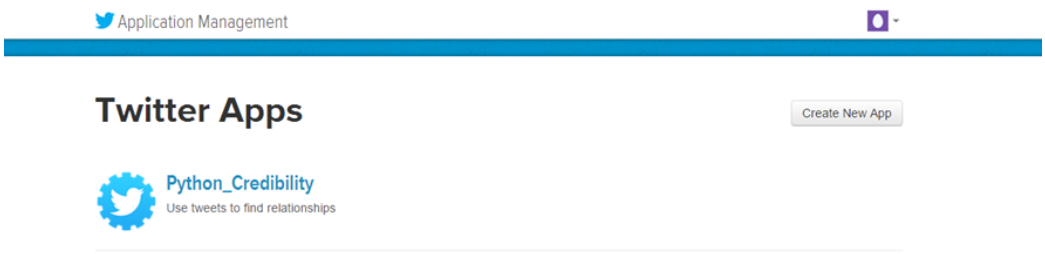

Figura 2.4: Interfaz de Generación de Apps

2.6 Conclusiones

Con la evolución de las nuevas tecnologías se ha propiciado un cambio dentro del modelo clásico de recolección de la información. En la actualidad gran parte de campo de la minería de datos, gira en torno a utilizar a las personas como fuentes de información.

Este cambio ha posicionado en una situación privilegiada como fuentes de información a las denominadas redes sociales, las cuales han propiciado un cambio importante en los modelos clásicos de transmisión de la información, donde se ha pasado de una estructura en la cual se diferenciaban claramente los generadores y los consumidores de contenido a una situación en la que esta diferenciación se convierte en muy difusa.

Este cambio ha sido impulsado en gran medida por la aparición de las redes sociales de seguimiento independiente favoreciendo que todos los usuarios, además de ser consumidores de contenidos; se conviertan en generadores de contenido.

Esto conlleva que la capacidad para contrastar información llegue a un punto que se transforma en complicada y, por tanto, sea necesaria la implementación de sistemas capaces de contrastar el contenido que se está transmitiendo. Con la finalidad de limitar el exceso de información y proporcionar al usuario la información creíble más relevante se convierte en necesaria la realización de una clasificación de los usuarios involucrados en un contexto en función de su grado de relevancia dentro del mismo.

Sin embargo, la relevancia dentro de los contextos no presenta ningún tipo de relación con la credibilidad del contenido del mensaje. Por ello es necesario establecer mecanismos capaces de clasificar el contenido. Sin embargo, dado que el análisis de esta credibilidad se debe de realizar sobre

textos cortos (se está trabajando sobre Twitter), no se pueden utilizar sistemas de clasificación basados en el análisis semántico de los textos, sino que se debe encontrar modelos que se basen en los metadatos que Twitter proporciona con cada tweet para poder discernir acerca de la credibilidad del contenido.

Es este el punto sobre el cual el programa *Sniffer* va a desarrollar su teoría y tratar de implementar una herramienta que ayude a la clasificación de la credibilidad de los contenidos agrupando a sus creadores en función de su grado de repercusión dentro de la red social Twitter.

Capítulo 3

Marco Legislativo del Proyecto

A lo largo de este capítulo se trata de exponer los límites de uso que se puede hacer con los datos procedentes de Twitter, enmarcándolos en la política de Twitter y a nivel español. El motivo de realizar este punto separado es que se van a utilizar datos de índole personal.

3.1 Introducción: Twitter a nivel Español

El proyecto que se va a desarrollar a lo largo del presente documento, tiene asociada la recolección de datos de índole personal: nombres de usuario, nombres, estadísticas de las cuentas o los mensajes que están emitiendo. Al tratarse de datos de carácter privado y personal es necesario enmarcar esta extracción de datos en la situación legislativa actual.

En España existe una legislación muy restrictiva en cuanto al uso de los datos personales de las personas. La principal legislación que existe en España es la Ley Orgánica 15/1999 del 13 de diciembre de Protección de Datos de Carácter Personal [16].

Esta ley impone diferentes obligaciones en el proceso de recogida de datos. En virtud del art. 6.1 *"El tratamiento de los datos de carácter personal requerirá el consentimiento inequívoco del afectado, salvo que la ley disponga otra cosa"*. Para que sea necesaria la obligación de cumplimiento de este artículo se debe de definir inequívocamente lo que se considera como dato de carácter personal. Esta definición se presenta en esta misma ley y se muestra a lo largo del art. 3 definiéndose como *"Datos de carácter personal: cualquier información concerniente a personas físicas identificadas o identificables"*

Según las resoluciones de la Agencia Española de Protección de Datos [17] (AEPD), los datos que extrae Twitter de los usuarios y posteriormente proporciona a los desarrolladores se pueden considerar como datos de carácter personal debido a que cumplen dos criterios establecidos:

- Permiten la identificación del usuario real.
- Son considerados datos.

El desarrollo de la autora lo relaciona con las direcciones de correo electrónico, en este artículo se establece lo siguiente: *"La dirección de correo electrónico tiene la consideración de dato de carácter personal porque, si bien el procesado de éstos no revela nuevas características referentes al comportamiento de las personas sí permite, lógicamente, su identificación"*.

Esto es fácilmente extrapolable a la red social Twitter donde a los desarrolladores se les proporcionan dos elementos identificativos, *"screen_name"* y *"user_id"*, cuando reciben un tweet. El primer elemento es único para el usuario en la red, aunque se pueda ver alterado con el paso del tiempo, se puede localizar e identificar al usuario con él. Por el contrario el segundo elemento no se modifica y se vincula de forma indefinida al usuario desde la creación de la cuenta por tanto el screen_name y el user_id que se recibe en la respuesta de la aplicación se consideran datos de carácter personal y por tanto se deben de regir por, en nuestro caso; la Ley Orgánica 15/1999 del 13 de diciembre de Protección de Datos de Carácter Personal.

Es por estos motivos por los que se tiene que tratar los datos que se recojan acorde con la legislación referente a la protección de datos de carácter personal. Es decir, no se puede publicar datos que puedan ser relacionados con el propietario, por lo que cuando se publiquen el user_id y el screen_name de cada usuario en la visualización se deberá de utilizar algún algoritmo que realice una transformación irreversible.

3.2 Información cedida a Twitter

Todos los usuarios que pretenden crearse una cuenta en la red social Twitter están obligados a aceptar la política de privacidad del servicio [18]. De entre estos términos aparece el más importante para nosotros, que dice lo siguiente:

"Esta licencia permite que demos acceso a todo el mundo a sus tweets en los Servicios de Twitter y que el resto de los usuarios puedan hacer lo

mismo."

Esto es la base de la red social, todos los usuarios pueden ver tu contenido pero a cambio tú puedes ver el contenido del resto. Es por ello que se deben delimitar cuáles son los datos que se van a utilizar. Si se continua analizando los términos de la política de privacidad de la red social, se informa de cuáles son los datos que se van a tratar.

Figura 3.1: Página de Registro de Twitter

Los datos que se van a tratar, y que nos informan los Servicios de Twitter cuando se aceptan los términos y condiciones del contrato; son:

- Información básica de la cuenta: se facilita el nombre, nombre de usuario, contraseña, correo electrónico o número de teléfono. Los servicios de Twitter se reservan el derecho a hacer públicos estos datos.
- Información del contacto: se facilita el correo electrónico o el número de teléfono. No haciéndose públicos estos datos pero los Servicios de Twitter pueden vincular distintos usuarios a través de estos datos.
- Información adicional: datos procedentes del perfil del usuario como descripción o sitios web pueden ser utilizados por los Servicios de Twitter y hacerse públicos.
- Tweets, gente que sigue, listas y otra información pública: con la finalidad de aportar una mejor experiencia los Servicios de Twitter pueden hacer pública este tipo de información.

Al analizar la política de privacidad de Twitter se nos informa de lo siguiente: *"Al utilizar cualquiera de nuestros Servicios, usted da su consentimiento para la recopilación, la transferencia, la manipulación, el*

almacenamiento, la revelación y otros usos de su información según lo descrito en esta Política de Privacidad"[18]

Los motivos que alegan los Servicios de Twitter para que esto se deba ceder son dos:

- Mejorar la experiencia del usuario: Sabiendo los gustos del usuario se le pueden recomendar ciertas cuentas, anuncios o incluso lugares de ocio dentro de la misma red social.
- Mantener el correcto funcionamiento de la plataforma: La información personal recopilada, descrita a continuación, se utiliza para proporcionar, supervisar y mejorar nuestros Servicios de manera continua.

En el momento en el que se utilizan datos de índole personal, se necesita del consentimiento del propietario de los datos (en este caso del usuario) para que se puedan utilizar. Twitter es una red social que como se ha visto necesita datos del usuario, por tanto es necesario dar el consentimiento de uso si se quiere utilizar la plataforma. Los Servicios de Twitter se escudan en que si no se acepta la cesión de la explotación de los datos no puede realizar el servicio para el que está destinado, por tanto le rechaza el acceso al mismo.

3.3 Información proporcionada por Twitter

Con la finalidad de extraer la información de Twitter es necesario el registro como desarrollador. Al realizarse este registro Twitter proporciona una serie de definiciones con el objetivo de obtener una experiencia en el uso de sus servicios por parte del desarrollador, estos servicios están expuestos en la política de desarrolladores de Twitter [19]. Entre lo que Twitter proporciona al desarrollador se encuentra lo siguiente:

- API de Twitter: es la interfaz de programación de aplicaciones de Twitter. Esta herramienta aporta la documentación necesaria para la extracción de contenido desde la red social y también la estructura con la que dichos datos se reciben, esta estructura se tratará más adelante.
- Contenido: Son los tweets, identificadores, información de perfiles y otros datos. Este contenido se recibe implementando las herramientas que se proporcionan en la API.
- Claves de acceso: En todo momento el acceso al producto de los Servicios de Twitter se debe de realizar identificándose, por ello es necesario la utilización de claves de usuario y de aplicación.

- Límites de uso: Debido a que es un servicio de código libre y se está tratando con información personal, los Servicios de Twitter limitan el acceso a su producto estableciendo típicamente límites de número de tweets por minuto y antigüedad.

3.4 Posibles usos de la información recibida

Cuando un usuario se registra como desarrollador la política de desarrolladores de Twitter [19] obliga, entre otras cosas; a comprometerse con lo siguiente:

- Conservar la integridad de los productos
- Respetar la privacidad del usuario
- Identificar el servicio
- No realizar SPAM
- No reproducir la experiencia de Twitter
- Limitar el uso comercial

Los tweets se pueden utilizar con fines comerciales, para poder realizarlo hay que informar a Twitter y abonarle una tarifa correspondiente al volumen de negocio. Hay que remarcar que no se prohíbe la visualización de los datos del usuario cuando se visualizan los tweets desde plataformas externas a Twitter.

Siempre que se vaya a mostrar el contenido íntegro del tweet se debe de realizar de una manera adecuada como se establece en los requisitos de visualización de su contenido [20] en caso de explotación con fines comerciales. Lo establecido dentro de estos requisitos son las definiciones de lo que significa cada contenido, esto es lo que se representa en la siguiente figura:

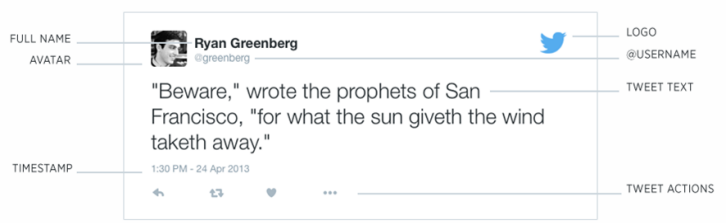

Figura 3.2: Formato de un tweet

No obstante, dado que la visualización de los datos no está relacionada con fines comerciales sino de académicos no estamos obligados a utilizar este formato de visualización. A lo que sí que estamos obligados es a disponer el logo de Twitter en la página en la que se vayan a representar los datos.

Capítulo 4

Estudio de los Metadatos

En este capítulo se tratara el estudio de las variables que se proporcionan con los metadatos del tweet procedente de Twitter. La finalidad de este estudio es obtener una base teórica entre las relaciones para la posterior implementación en el programa "Sniffer".

4.1 Introducción

Se ha tratado de explicar a lo largo del capítulo relacionado con el estado del arte, la situación más óptima para la realización de análisis de información cuyo contenido se presenta en forma de texto con una longitud reducida es basarse en los metadatos que dicho texto tenga asociados. Para el caso que concierne a este proyecto, como se ha informado anteriormente; los Servicios de Twitter proporcionan una serie de metadatos en cada una de las consultas, el volumen de información es suficiente como para poder plantear un análisis de los mismos.

Durante este capítulo se realizará un estudio en relación con estas variables. El principal objetivo de este estudio de las variables es obtener la estructura de relaciones que se debe utilizar en el sistema que se desarrollará a lo largo de los siguientes capítulos. Este programa es al que se refiere este documento como *Sniffer*. El principal objetivo de este programa es elaborar un sistema que permita ayudar a clasificar, de forma automática, la credibilidad de diferentes tweets enmarcados en un contexto.

Como es lógico, previo paso a desarrollar el sistema, se debe entender el entorno en el que se va a trabajar, esto involucra el tener que estudiar la relación que tienen las variables y el poder de influencia que poseen en

el momento de la decisión. Este capítulo pretende mostrar el proceso de análisis y simulación del sistema final.

Para la realización de este estudio se han utilizado diferentes herramientas. Estas herramientas son sistemas dedicados al tratamiento de datos y ayudan a la comprensión de las relaciones entre las distintas variables de cara a la extracción de información necesaria para la clasificación. Las herramientas que han sido utilizadas para el análisis de los datos han sido:

- Watson Analytics
- Rapidminer

Estas herramientas están descritas en el capítulo correspondiente al desarrollo de las tecnologías utilizadas (apéndice D).

4.1.1 Objetivos

El principal objetivo que se va a tratar en este apartado es obtener una base para poder establecer las relaciones entre las variables que intervienen en la decisión del grado de credibilidad que tiene asociado un tweet. También se va a tratar un proceso de clasificación de usuarios con la finalidad de focalizar el entrenamiento de los clasificadores a su tipo de usuarios, con esto lo que se pretende obtener es una mejora en las prestaciones de los clasificadores para tratar de proporcionarles conjuntos lo más homogéneos posibles.

Por último se va a realizar la simulación de cada una de las fases de clasificación de las que va a constar el programa *Sniffer*. Con esto se pretende extraer una base sólida de la que partir en el proceso de programación del sistema final. Además se gana en velocidad de modificación de las estructuras de clasificación con respecto a si se hiciese directamente programando el sistema final.

Las fases en las que se va a dividir el estudio van a ser:

- **Segmentación de los usuarios**: En esta fase se va a tratar de extraer los conjuntos de usuarios lo más homogéneos posible.
- **Primer nivel de clasificación**: En esta fase se va a realizar caracterización de la primera clasificación de los tweets. En este primer nivel se pretende discernir en torno al contenido del mensaje, centrándose principalmente en la presencia o no de información y su relación con el contexto.

- **Segundo nivel de clasificación**: En esta fase se va a realizar la caracterización de la segunda clasificación de los tweets. En este segundo nivel es el encargado de aportar el grado de credibilidad a los mensajes que llegan hasta él, únicamente los clasificados como *Relacionados y con información*.

Los algoritmos que se vayan a utilizar en el programa en este capítulo no se van a tratar. En este capítulo se pretende realizar un análisis teórico del planteamiento que se va a realizar en el programa *Sniffer*. Este desarrollo se realizará en el capítulo correspondiente (Capítulo 5).

4.2 Procedimientos

Con la finalidad de realizar el estudio, se ha generado un conjunto de muestra con tweets de situaciones reales. La clasificación de los tweets se ha realizado por medio de una funcionalidad del programa *Sniffer* para la clasificación de los tweets, funcionalidad que se explicará en su capítulo correspondiente. Esta clasificación se ha realizado de forma manual, a la hora de la clasificación el usuario únicamente dispone de la siguiente información:

- **Nombre del usuario**: es el "@nickname" del usuario en Twitter
- **Texto del tweet**: es el contenido que se está transmitiendo.
- **Clase**: esta información aparece en la versión final de la clasificación de los tweets. Es información orientativa de cómo ha realizado la primera clasificación el sistema para ese tweet.
- **Creíble**: esta información aparece en la versión final de la clasificación de los tweets. Es información orientativa de como se ha realizado la segunda clasificación del sistema para ese tweet.

El proceso de clasificación de la credibilidad que se plantea está estructurado por medio de la secuencia de ejecución representada en la figura 4.1.

Como procedimiento de análisis de los datos se va a realizar una selección de las variables, y un posterior estudio acerca de su implementación mediante algoritmos de clasificación basados en aprendizaje. Un apartado que no se va a tratar en este capítulo va a ser lo relacionado con la obtención de altavoces. El motivo de esto es que los altavoces no influyen en los parámetros de clasificación, simplemente se plantea esta obtención con fines de separar mejor la presentación de los resultados, con la finalidad de que el usuario pueda centrarse únicamente en los tweets con una mayor repercusión.

Figura 4.1: Secuencia de Clasificación

4.3 Clasificación de los Usuarios

Un problema al que nos enfrentamos en Twitter es la enorme variedad que existen entre sus usuarios. Twitter, como se ha comentado, constituye un entorno donde las relaciones de seguimiento son independientes. Esto está propiciando la aparición de diferentes clases de cuentas.

Existen infinidad de clasificaciones de usuarios de Twitter, atendiendo a la ratio seguidores/seguidos, al número de tweets diarios... La clasificación final que se quiere realizar en Twitter tiene como base conseguir distinguir los tipos de comportamientos en torno al estilo de comunicación de los diferentes usuarios de la red social. Existen diferentes políticas de comunicación: basar la comunicación en la interacción con los usuarios, realizar únicamente difusión de información. Es por este motivo por el que se plantean los siguientes tipos de usuarios:

- **Usuarios normales**: Basan su política de comunicación en la interacción con el resto de usuarios, su información procede de diferentes fuentes. El perfil asociado a este tipo de usuarios son las personas físicas por lo que la red de difusión que se establece en torno a ellos suele ser muy limitada.

- **Usuarios corporativos**: Basan su política de comunicación en la difusión de información, sin embargo, poseen un cierto grado de interacción con los usuarios. El perfil que abunda en este tipo de usuarios está compuesto por medianas empresas por lo que el poder de difusión esta ciertamente limitado pero es bastante superior al de los usuarios normales.

- **Usuarios organizativos**: Su política de comunicación se basa únicamente en la difusión de información. El perfil asociado a este tipo de usuarios suele estar ligado a organismos estatales o grandes empresas,

por lo que su poder de difusión es muy elevado y, por tanto, el alcance de sus publicaciones suele ser muy grande.

Esta clasificación representa una de las principales bases del proyecto. No se puede considerar que en una red social todos los usuarios se comportan de la misma manera. Si se hiciese esta consideración en un sistema de clasificación, se estaría perjudicando a las prestaciones de los diferentes clasificadores que se dispongan en el sistema.

Con la finalidad de conseguir esta clasificación, se puede plantear que el factor determinante para la clasificación dependa del número de usuarios seguidos y del número de seguidores que un usuario tenga. Esta variable para clasificar a los usuarios no presenta otro motivo que los siguientes:

- Las cuentas de un usuario medio suele tener una proporción entre los usuarios seguidos (*friends*) y los seguidores (*followers*) inferior a 1, no obstante en ciertas ocasiones esta proporción suele ser mayor por lo que se puede utilizar un baremo un poco superior a este valor.

- Las cuentas de los usuarios que representan a empresas, suelen tener una relación bastante superior a 1, es decir, poseen más seguidores que seguidos. Por tanto, si se quiere enmarcar a estos usuarios, se debe establecer un filtro que abarque unos rangos superiores.

- Por ultimo las cuentas de alta repercusión, medios informativos, organismos institucionales... suelen presentar un número muy pequeño de seguidores en comparación con los usuarios que les siguen, es por ello por lo que se les puede situar en un filtro que sea muy superior a los anteriores.

Para realizar el estudio se ha utilizado el total de los usuarios que la aplicación *Sniffer* ha registrado durante el periodo de captación de tweets. En total son 65.535 registros de usuarios los cuales van a permitir establecer los filtros para poder separarlos en subgrupos.

El conjunto inicial de partida es el que se puede ver en la figura 4.2. Este histograma está limitado a que muestre únicamente los usuarios que tienen una ratio seguidores/seguidos menor que cien, el motivo de esto es simplemente por motivos de visualización, no se puede elaborar un histograma que se vea fácilmente la distribución si se tiene usuarios con ratios de hasta 100.000.

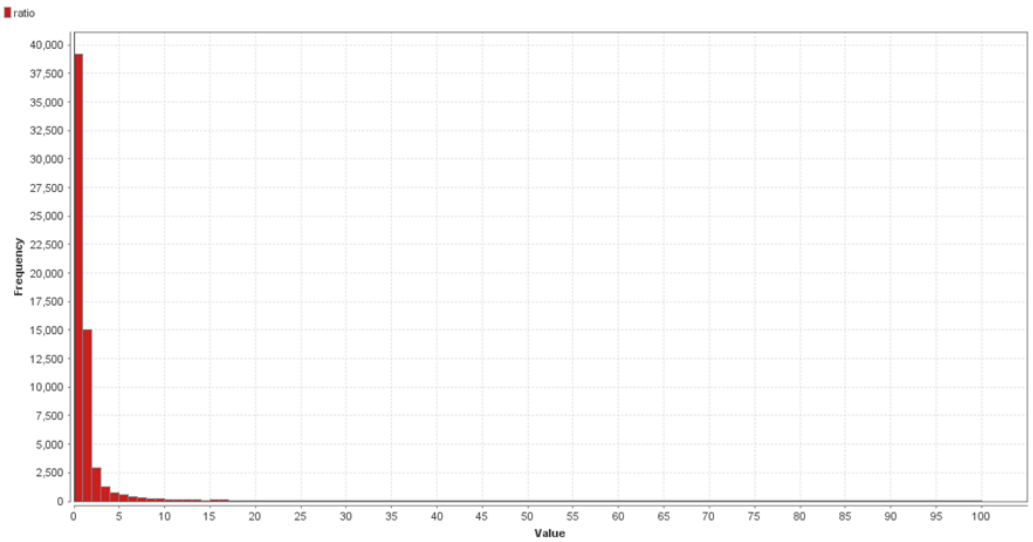

Figura 4.2: Histograma de los Ratios de los Usuarios

Como puede verse, de los 65535 usuarios que se han tomado, más de 52.500 usuarios (alrededor del 80 %) poseen una ratio inferior a 2, estos son los que se han denominado como usuarios normales. Se puede ver que existen muestras de usuarios que poseen ratios superiores a 2 pero inferiores a 10, representan un 11 % (alrededor de 7.500 usuarios) del conjunto total por lo que se podría llegar a limitar entre esos dos valores.

Finalmente se puede observar que a partir de un ratio superior a 10 el total de los usuarios no representa más del 9 % (alrededor de 5.000 usuarios). Esto hace ver que se están estableciendo unos criterios de selección que se adaptan correctamente a la situación de Twitter.

Por tanto, para el proceso de clasificación de los usuarios se utilizarían los tres siguientes filtros:

- **Usuarios Normales**: Son todos los usuarios que posean una ratio de seguidores/seguidos inferior a 2.
- **Usuarios Corporativos**: Son todos los usuarios que tengan comprendida la ratio de seguidores/seguidos entre 2 y 10.
- **Usuarios Organizativos**: Son todos los usuarios que posean una ratio de seguidores/seguidos superior a 10.

A continuación a este análisis se ha elaborado una simulación de este

proceso por medio de la herramienta RapidMiner. En esta herramienta se han dispuesto los elementos necesarios para la clasificación de acuerdo a lo anteriormente explicado. El esquema del proceso está representado en la figura 4.3 y los resultados figuran en la figura 4.4.

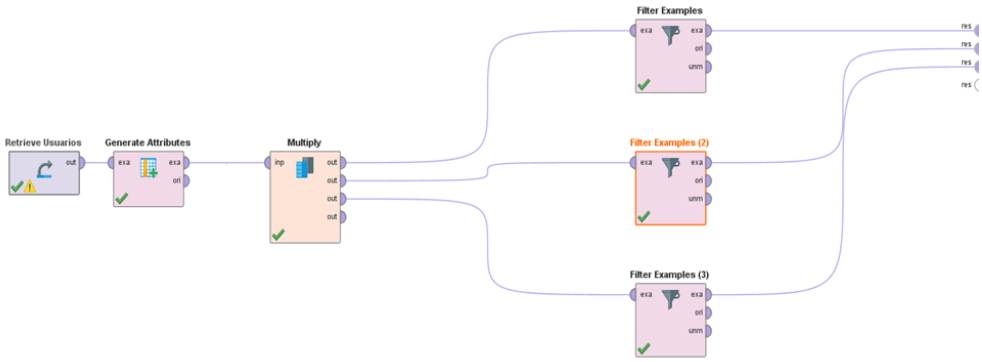

Figura 4.3: Representacion del proceso en RapidMiner

Este proceso será el primer paso que se haga en el programa *Sniffer* por los motivos que se han tratado de exponer a lo largo de éste apartado. Con esto se ha conseguido concretar el rango de actuación que tendrá cada clasificador dentro del sistema.

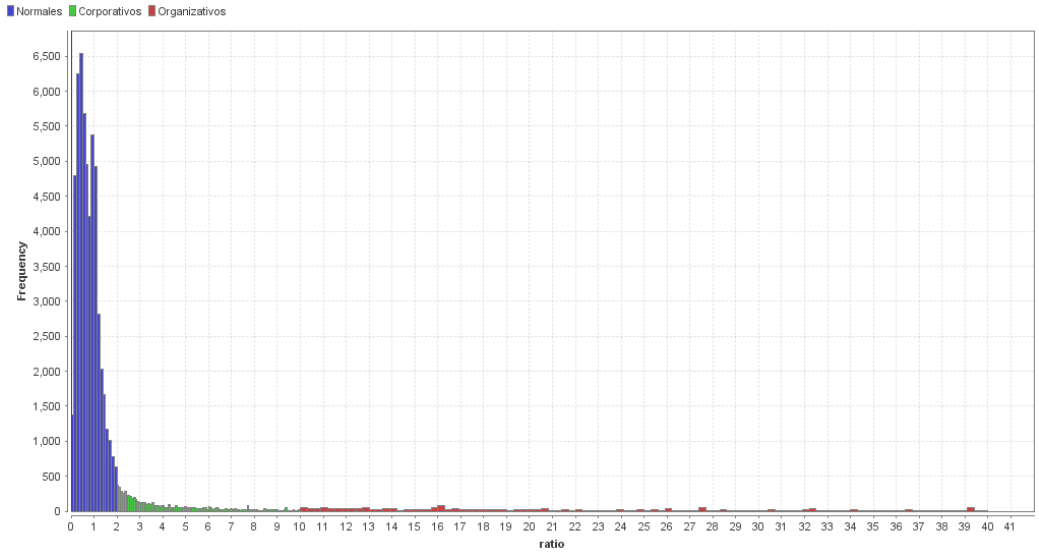

Figura 4.4: Resultados de la Clasificación

4.4 Primer Nivel de Clasificación

A lo largo de este segundo apartado se va a tratar la clasificación de los contenidos de los tweets. Como es de todos sabido, cuando se está tratando con un conjunto de datos pueden aparecer ciertos registros que no guarden relación o que no transmitan información. En este primer nivel de clasificación se va a tratar de realizar esta primera criba de datos.

Cuando se trata de analizar la credibilidad, únicamente interesan los contenidos que están relacionados con el tema y que además transmiten información, es por ello por lo que este primer nivel extraerá cuáles son los tweets que transmiten información para que continúen con las siguientes fases de clasificación. Por tanto, este nivel de clasificación reparte a los tweets en 4 categorías distintas:

- **Tweets Relacionados con Información** (R1): Estos son los tweets que interesa extraer su credibilidad. Son tweets que están relacionados y que además transmiten cierta información.
- **Tweets Relacionados sin Información** (R2): Estos tweets podrían interesar, sin embargo no tienen ninguna información sobre el contexto, pese a estar relacionados con él.
- **Tweets no Relacionados** (R3): Dado que se está tratando con contextos concretos, estos tweets no son útiles pues no están relacionados con el tema que se está analizando.
- **Tweets Restantes** (R4): Dado que en el proceso de captura de tweets pueden aparecer errores, este nivel será al que se les adjudicará a errores o retweets.

De entre todas las variables que proporcionan los Servicios de Twitter, se ha realizado una selección de las mismas quedándose para realizar el estudio, las variables expuestas en la tabla 4.1.

La selección de estas primeras variables ha sido obtenida de los artículos [10] y [9]. Sin embargo, para la clasificación y la selección de los algoritmos se va a utilizar diferentes algoritmos a los planteados por los distintos autores.

Variable	Descripción
Longitud	Longitud del tweet en caracteres
Número de palabras	Número de palabras que posee el tweet
Número de monosílabos	Número de palabras monosilábicas que posee el tweet
Número de vía	Número de *vía* que contiene el tweet
Número de interrogaciones	Número de marcas de interrogación que contiene el tweet
Número de exclamaciones	Número de marcas de exclamación que contiene el tweet
Edad del tweet	Momento en el que se publicó el tweet respecto del inicio del contexto (en horas)
Número de URL's	Número de URL que contiene el tweet
Número de menciones	Número de menciones de usuario que contiene el tweet
Número de hashtags	Número de hashtags que contiene el tweet
Posee localización	Si el tweet posee localización
Ratio Tweets/Seguidores	Relación entre el número de tweets publicados y el número de seguidores
Ratio Seguidos/Seguidores	Relación entre el número de seguidos y el número de seguidores
Usuario Verificado	Si el usuario esta verificado o no

Tabla 4.1: Variables influyentes en la Clasificación

A partir de los metadatos que se han extraído de los tweets se han obtenido cada una de estas variables registrándose en un fichero con la finalidad de poder tener almacenados los datos para el siguiente estudio. En primer lugar se ha hecho uso de la herramienta de Watson Analytics[1]. La utilidad de esta herramienta ha sido que ha permitido obtener cuales son las variables más influyentes en el proceso de selección de los parámetros. Para ello hace uso de un árbol de decisión (figura 4.5). En este árbol de decisión establece que las siguientes reglas son las que presentan un mayor poder de clasificación:

- ¿Posee URL?
- ¿Cuál es la edad del tweet?

Estas reglas parecen razonables por los siguientes motivos: en 140 caracteres no es posible exponer toda la situación o toda la información relacionada por lo que si se añade una URL que vincula con un elemento externo o alguna prueba gráfica se puede transmitir una mayor información. Por otro lado conforme más edad tienen los tweets respecto del inicio del contexto la información es mayor pues hay más fuentes transmitiendo y se ha podido haber recabado y contrastado información procedente de diferentes fuentes.

[1]http://www.ibm.com/analytics/watson-analytics/us-en/

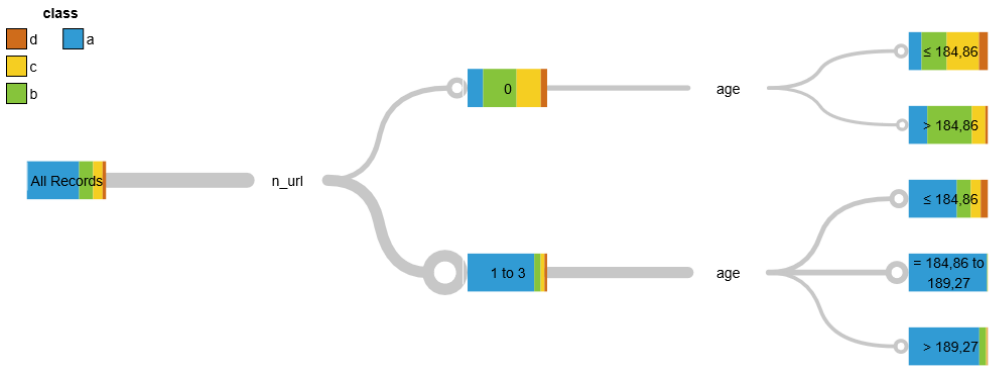

Figura 4.5: Árbol de Decisión

Para comprobar cuáles son las prestaciones en la simulación en el modelo planteado, se dispone un sistema similar sobre RapidMiner con las reglas que ha dictaminado la herramienta Watson Analytics. Sin embargo, estas reglas se quedan un poco escasas a la hora de realizar la clasificación de los tweets, pues realizando las simulaciones se arrojan los resultados de la tabla 4.2.

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Relacionados con información	84,57 % (63)	70,00 % (20)	88,46 % (29)
Relacionados sin información	53,12 % (32)	0,00 % (0)	57,20 % (15)
No relacionados	100 % (1)	66,67 % (3)	0,00 % (3)
Resto	0,00 % (1)	0,00 % (0)	0,00 % (0)
Resolución	72,16 %	69,57 %	79,32 %

Tabla 4.2: Resultados de precisión (número de muestras) de la primera clasificación (Árbol de Decisión).

Esta tabla pretende representar las prestaciones, en cuanto a términos de precisión se refiere, que es posible obtener para cada una de las clases del nivel de clasificación. Como puede verse están disgregados los datos en tres grupos de usuarios, estos grupos son los resultantes de la clasificación expuesta en el apartado anterior.

Como puede observarse en los resultados, existe un posible margen de mejora, es por ello por lo que se va a tratar de comprobar cómo se desenvuelven otros algoritmos de clasificación para este caso. Los algoritmos que se pretende comprobar sus prestaciones pertenecen a la familia de los algoritmos de aprendizajes basados en inducción supervisada. De entre los métodos existentes de los que se evaluará su funcionamiento son los siguientes:

- KNN (K-Nearest Neighbors): Este algoritmo presenta dos variantes que se analizarán: emplear una distancia euclídea y emplear una distancia de Mahalanobis. La principal ventaja de este algoritmo es su rápido entrenamiento lo que se ve contrarrestado con el tiempo necesario para la clasificación.

- Random Forest: Es uno de los algoritmos que mejor aproxima al resultado verdadero, sin embargo existe una facilidad para el sobreajuste en conjuntos de datos muy ruidosos.

- Perceptrones Multicapa (MLP): Este algoritmo está basado en redes neuronales y compone un subconjunto de las redes.

Se comienza el desarrollo de la simulación con el algoritmo *KNN*, hay que destacar que únicamente se puede probar el funcionamiento de la simulación con la distancia euclídea pues la distancia de mahalanobis RapidMiner no la tiene implementada. A diferencia de los árboles de decisión en esta simulación se están utilizando todas las variables salvo verificado y número de vía. Estableciéndose la separación previa de los usuarios, como ya se había hecho con el árbol de decisión; se obtienen los resultados presentes en la tabla 4.3:

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Relacionados con información	73,24 % (71)	78,26 %(23)	89,66 % (29)
Relacionados sin información	43,75 % (16)	100 % (1)	0 % (1)
No relacionados	20,00 % (10)	0 % (0)	100 % (1)
Resto	0 % (0)	0 % (0)	0 % (0)
Resolución	62,89 %	79,15 %	89,66 %

Tabla 4.3: Resultados de precisión (número de muestras) de la primera clasificación (KNN)

El motivo de la utilización de esta cantidad de variables en estos algoritmos es debido a que, a diferencia de los árboles de decisión, estos algoritmos necesitan un mayor número de variables para poder determinar la clase a la que pertenece cada mensaje. Los árboles de decisión al realizar una clasificación en función de la mayor influencia necesitan una menor cantidad de variables para discernir, pues aplican el denominado *"podado"* de las ramas que presentan pocos elementos.

Con esta simulación se puede ver que el algoritmo KNN está siendo peor para las muestras pertenecientes a los usuarios normales, sin embargo mejora el funcionamiento claramente en los usuarios corporativos y sobretodo en usuarios organizativos respecto de la clasificación basada en el árbol de decisión. Con estos resultados, se podría llegar a plantear la implantación de la presencia de ambos clasificadores trabajando de forma paralela en el sistema para que uno supla las carencias del otro y viceversa.

Se prosigue con el análisis de los distintos algoritmos con el clasificador *Random Forest*, como ya se ha dicho este algoritmo puede ser muy preciso pero se necesita tener mucho cuidado en no caer en sobreajuste. Para este algoritmo de entrenamiento se han seleccionado las mismas variables que para los KNN. Los resultados que ha arrojado la simulación utilizando este algoritmo han sido los representados en la tabla 4.4.

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Relacionados con información	77,14 % (70)	78,26 %(23)	89,66 % (29)
Relacionados sin información	56,52 % (23)	0 % (0)	0 % (1)
No relacionados	0 % (3)	0 % (0)	100 % (1)
Resto	0 % (1)	0 % (0)	0 % (0)
Resolución	69,07 %	78,26 %	89,66 %

Tabla 4.4: Resultados de precisión (número de muestras) de la primera clasificación (Random Forest)

Se puede seguir viendo que la clasificación que realiza el algoritmo basado en árboles de decisión tiene más problemas que el KNN y el *Random Forest*, no obstante, estos dos últimos algoritmos mejoran mucho las prestaciones en usuarios corporativos y organizativos, por lo que sigue siendo interesante

ver como se desenvuelven estos 3 algoritmos trabajando a la par.

Por último se va a proceder a analizar las prestaciones que pueden obtenerse utilizando el algoritmo de clasificación MLP, este algoritmo al igual que ocurre con el Random Forest tiende a establecer un sobreajuste fácilmente. Sin embargo se desenvuelve perfectamente en entornos muy ruidosos por lo que es una opción que puede llegar a ser muy interesante. Los resultados arrojados por esta simulación han sido los representados en la tabla 4.5.

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Relacionados con información	78,69 % (61)	77,27 %(22)	92,86 % (28)
Relacionados sin información	44,44 % (27)	0 % (1)	0 % (0)
No relacionados	33,33 % (9)	0 % (0)	100 % (1)
Resto	0 % (0)	0 % (0)	0 % (0)
Resolución	64,95 %	73,91 %	93,10 %

Tabla 4.5: Resultados de precisión (número de muestras) de la primera clasificación (MLP)

A simple vista, se puede observar que el sistema basado en MLP tiene su máximo uso para el conjunto de los usuarios organizativos. Cabe de entender que en el momento en el que los clasificadores interactúen todos en común el MLP tendrá una mayor capacidad de decisión final en el grupo de usuarios organizativos.

La estructura que se ha utilizado en la implementación de las distintas simulaciones en RapidMiner ha sido la presente en la figura 4.6, los únicos bloques que se han variado para las diferentes simulaciones han sido:

- Selección de atributos (Select Attributes en la imagen): Dado que los atributos que se utilizan en el árbol de decisión son distintos del resto, se ha tenido que modificar estos parámetros en la decisión.
- Clasificador (Decision Tree en la imagen): Constituye el algoritmo de clasificación que se va utilizar en cada caso.

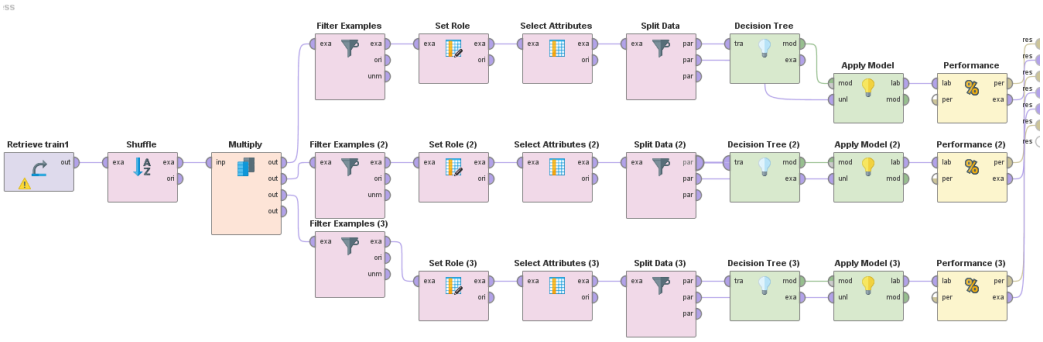

Figura 4.6: Diagrama de Bloques de la Simulación

Una vez estudiados por separado las diferentes opciones de algoritmos de clasificación que se tienen disponibles, se procede a ver cómo funcionan todos trabajando en conjunto. El principal motivo que lleva a realizar un sistema de estas características es que la precisión de los clasificadores no es uniforme para todos los conjuntos, por ello se pretende comprobar cómo se desenvuelven en conjunto para ver si se pueden suplir las carencias de unos con los otros. Se va a plantear un sistema como el presente en la figura 4.7 y se realizará la simulación por medio de la herramienta RapidMiner.

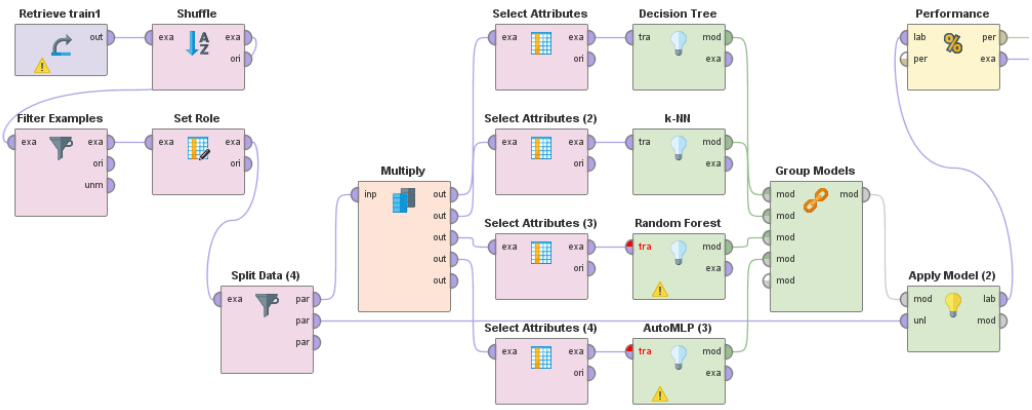

Figura 4.7: Esquema Primera Clasificación

Las simulaciones se han realizado para los tres conjuntos de datos, variándose para ello las características del filtro empleado. Los resultados que han arrojado estas simulaciones han sido los representados a lo largo de la tabla 4.6.

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Relacionados con información	74,32 % (74)	80,95 %(21)	91,30 % (23)
Relacionados sin información	80,00 % (5)	0 % (0)	28,57 % (7)
No relacionados	44,44 % (18)	100 % (2)	100 % (1)
Resto	100 % (1)	0 % (0)	0 % (0)
Resolución	70,95 %	82,61 %	78,56 %

Tabla 4.6: Resultados de precisión (número de muestras) de la primera clasificación (Conjunto)

Por tanto si se pone en comparación a todos los clasificadores (tabla 4.7) se tiene que el clasificador con mayor precisión es el que emplea el algoritmo de aprendizaje *Random Forest*. Sin embargo, presenta una variación muy grande entre las distintas clasificaciones de usuarios. Por otro lado, la precisión del conjunto de decisores es la que presenta una mayor precisión y además, no presenta tanta variación entre las distintas clasificaciones. Por tanto, se puede considerar que para esta primera clasificación se puede utilizar un sistema de clasificación basado en el conjunto de algoritmos desarrollados[2].

Algoritmo	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**	**Precisión Media**
Árbol de decisión	72,16 %	69,57 %	71,42 %	71,05 %
KNN	62,89 %	79,15 %	89,66 %	77,23 %
Random Forest	69,07 %	78,26 %	89,66 %	78,99 %
MLP	64,95 %	73,91 %	93,10 %	77,32 %
Conjunto	70,95 %	82,61 %	78,56 %	77,38 %

Tabla 4.7: Comparativa de Prestaciones de la Primera Clasificación

[2]Todos los parámetros de cada clasificador se han dispuesto en el Capítulo 5

Para resumir por tanto este apartado, el sistema correspondiente a la primera clasificación se encargará de dictaminar la relación del mensaje con el contexto y comprobar si contiene información. Para ello se hará un uso de sistemas de clasificación basados en algoritmos de aprendizaje automático. La estructura del sistema de clasificación constará de cuatro clasificadores (Árbol de decisión, KNN, *Random Forest* y MLP) pues presenta un poder clasificatorio más estable frente a los tres tipos de usuarios.

4.5 Segundo Nivel de Clasificación

A lo largo de este apartado se va a tratar el estudio de las variables que influyen en la toma de decisiones correspondientes al segundo nivel de clasificación del sistema *Sniffer*. A este segundo nivel de clasificación, únicamente acceden los tweets que han sido clasificados como R1 es decir relacionados con el contexto y poseedores de información.

La obtención de los altavoces tampoco presenta un carácter influyente a la hora de clasificar los tweets. Como se ha comentado, esta obtención y separación en función de su relevancia se realiza únicamente con carácter estructural y con la finalidad de facilitar la comprensión de los resultados.

Así pues, a lo largo de este último nivel es donde finalmente se realiza la clasificación de los tweets en función de su credibilidad. Se puede considerar que hasta este punto lo que únicamente llega es información pura, se han eliminado los retweets, todo lo no relacionado y todo lo que no contenía información. A lo largo de este apartado la clasificación en la que se va a establecer el grado de credibilidad de los tweets se compone de las siguientes clases:

- **Creíble** (C1): Estos tweets son los que mayor grado de credibilidad poseen. Son al fin y al cabo los que presentan una mayor prioridad para clasificarlos, pues son los más útiles de cara al transmitir información en el contexto.
- **Poco creíble** (C2): Estos tweets se componen de un grado de credibilidad inferior a los de C1. La información que estos tweets transmiten ha de ser interpretada con cuidado por parte del usuario, pues no se garantiza su total credibilidad.
- **No creíble** (C3): Estos tweets se pueden considerar que no tienen una base creíble dentro del contexto. Es por ello por lo que, en la medida de lo posible; se deben de descartar en la toma de datos.

Los sistemas que se van a crear para dicho propósito se van a basar en las mismas variables en las que se ha basado la primera clasificación. De entre todas estas variables, la herramienta Watson Analytics proporciona que las principales relaciones entre la clasificación y las variables iniciales son las siguientes (figura 4.8):

- El 75 % de los tweets creíbles poseían más de una URL's
- El 22 % de los tweets sin localización del usuario con una o menos URL's y con una relación Tweets/Seguidores inferior a 100 no son creíbles.

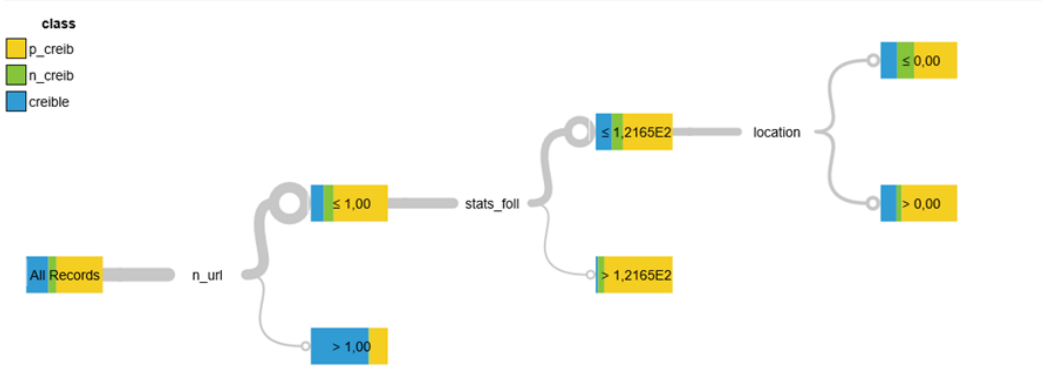

Figura 4.8: Árbol de decisión Segunda Clasificación

Es decir que al final las variables más importantes que van a permitir clasificar en tweet van a ser:

- Número de URL's
- Posee localización
- Relación Tweets/Seguidores

Como se ha realizado en el análisis de la primera clasificación se va a plantear como algoritmo de clasificación un árbol de decisión de acuerdo a los parámetros que ha arrojado Watson Analytics. Para la simulación la estructura que se ha seguido ha sido la misma que en el caso de la primera clasificación (figura 4.6) y los resultados que se han obtenido por medio de RapidMiner han sido los representados en la tabla 4.8.

A la vista de este resultado se puede observar que se presentan complicaciones a la hora de clasificar a los usuarios corporativos. El nivel de clasificación C2 parece que funciona muy bien y el nivel C1 funciona

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Creíble	100 % (3)	66,67 % (6)	86,67 % (15)
Poco Creíble	85,11 % (47)	81,82 % (11)	66,67 % (9)
No Creíble	100 % (5)	0 % (0)	0 % (0)
Resolución	87,27 %	76,47 %	79,17 %

Tabla 4.8: Resultados de precisión (número de muestras) de la segunda clasificación (Árbol de Decisión)

relativamente bien en usuarios normales. Relativamente porque hay que destacar el número de muestras que se han utilizado para la evaluación del nivel C1 para ese caso concreto. Por ello se deben de leer estos datos con cuidado. Este algoritmo de clasificación presenta unos resultados muy buenos, no obstante como ocurría en el caso anterior se entiende que pueden mejorarse. Es por ello por lo que se pretende probar los mismos algoritmos que se han probado en la primera clasificación para comprobar si es posible obtener unos mejores resultados de cara a la clasificación aunque se convierta en más complicado el sistema.

En consecuencia se va a comenzar, como en el caso anterior, evaluando el rendimiento que aporta el algoritmo basado en KNN empleando la distancia euclídea. El principal problema al que se enfrenta el planteamiento cuando se utiliza un KNN con distancia euclídea es que la variación en unos campos no representa lo mismo que en otros. Este sistema funciona muy bien para variables binarias, no obstante los resultados procedentes de la simulación han sido los representados en la tabla 4.9.

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Creíble	50,00 % (4)	50,00 % (4)	82,35 % (17)
Poco Creíble	77,55 % (49)	69,23 % (13)	71,43 % (6)
No Creíble	50,00 % (2)	0 % (0)	0 % (0)
Resolución	74,55 %	64,71 %	79,17 %

Tabla 4.9: Resultados de precisión (número de muestras) de la segunda clasificación (KNN)

Pese a que el número de muestras que se utilizan es reducido el KNN está arrojando unos resultados que son muy pobres para el sistema, únicamente es destacable la forma en la que se desenvuelve en entornos con usuarios organizativos en el que parece que iguala las prestaciones del árbol de decisión. Las variables que se han utilizado en este clasificador no son las mismas que para el árbol de decisión, sino que añade el número de hashtags y el número de menciones que contiene el tweet.

Continuando con el estudio de los diferentes clasificadores se va a implementar ahora el clasificador basado en el algoritmo Random Forest. La simulación del sistema de clasificación basado en este algoritmo retorna un rendimiento que está recogido en la tabla 4.10:

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Creíble	0 % (0)	50,00 % (2)	73,33 % (15)
Poco Creíble	78,43 % (51)	73,33 % (15)	83,33 % (6)
No Creíble	100 % (4)	0 % (0)	0 % (3)
Resolución	80,00 %	70,59 %	66,67 %

Tabla 4.10: Resultados de precisión (número de muestras) de la segunda clasificación (Random Forest)

A diferencia del primer nivel de clasificación el algoritmo Random Forest presenta unas prestaciones absolutamente inferiores en todos los tipos de clasificador al árbol de decisión por tanto se debería plantear no introducirle en el sistema final para este nivel de clasificación. Las variables para las que se obtiene el mejor rendimiento son para las mismas que posee el algoritmo basado en el árbol de decisión.

Para finalizar con el estudio de los diferentes algoritmos de clasificación cuando trabajan de forma independiente, se va a proceder a la simulación del sistema de clasificación basado en el algoritmo de MLP. Los resultados que proporciona la simulación son los representados en la tabla 4.11.

Los resultados que proporciona el algoritmo MLP hacen ver que, este sistema no es válido para este nivel de clasificación. Es, de todos los algoritmos simulados, el que peores prestaciones aporta a la hora de clasificar la credibilidad.

Como se ha podido ver ningún sistema ha sido capaz de superar o

Clase	Usuarios Normales	Usuarios Corporativos	Usuarios Organizativos
Creíble	0 % (0)	60,00 % (4)	78,57 % (14)
Poco Creíble	74,51 % (51)	77,78 % (9)	66,67 % (6)
No Creíble	50,00 % (4)	0 % (3)	25,00 % (4)
Resolución	72,73 %	58,82 %	66,67 %

Tabla 4.11: Resultados de precisión (número de muestras) de la segunda clasificación (MLP)

mejorar las prestaciones que aporta el algoritmo de árboles de decisión. Por tanto la opción de poder combinar la acción de varios clasificadores para la decisión se debe de descartar.

Una característica que dice que el sistema funciona bien es que, a la hora de clasificar en caso de error el sistema de clasificación asigna un nivel adyacente al real, es decir; cuando es *"creíble"* y se equivoca no asigna un grado *"no creíble"* sino un grado *"poco creíble"*. Esto ocurre debido el umbral de decisión es difuso y por ello considera error a estos casos, siendo un error de menor grado. Un error que no se debe cometer a este nivel, y que no se comete; es asignar *"creíbles"* a los *"no creíbles"* y viceversa. Finalmente el esquema que se va a plantear para este sistema de clasificación va a ser el representado en la figura 4.9:

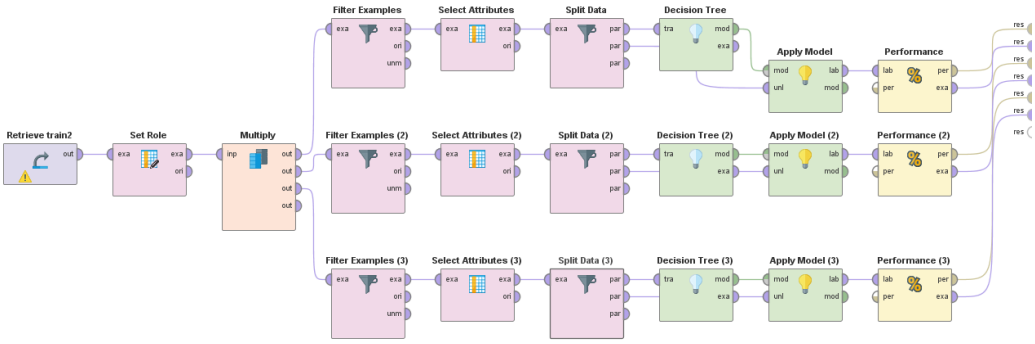

Figura 4.9: Esquema Segunda Clasificación (Montaje final)

Para resumir por tanto este apartado, el sistema correspondiente a la segunda clasificación se encargará de dictaminar el grado de credibilidad del mensaje con el contexto. Para ello se hará un uso de sistemas de clasificación basados en algoritmos de aprendizaje automático. La estructura del sistema

de clasificación constará de un único clasificador (Árbol de decisión) pues presenta un poder clasificatorio que no es capaz de mejorar o estabilizar ningún otro tipo de algoritmo de clasificación disponible frente a los tres tipos de usuarios.

4.6 Simulación Completa

Finalmente en este apartado lo que se pretende obtener es la presentación final de la implementación del sistema de clasificación. Cabe de destacar que este sistema que se va a introducir no tiene la presentación de los altavoces que se ha hablado en ciertos puntos a lo largo de este capítulo. La simulación por motivos de claridad del esquema se va a realizar en función de cada grupo de usuarios, es decir el sistema que se va a representar está diseñado para obtener las características de un grupo de usuarios. El esquema final que seguiría cada tweet es el que está representado en la figura 4.10.

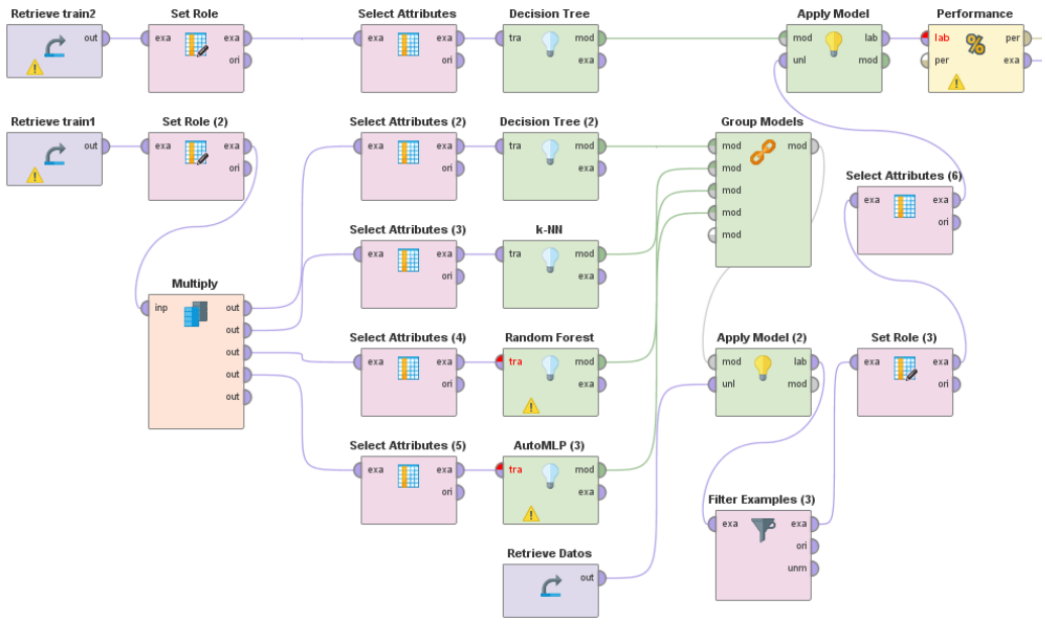

Figura 4.10: Representación del Esquema Final de Clasificación

Los resultados que se han obtenido finalmente con la simulación del proceso de clasificación han sido los mostrados a lo largo de la tabla 4.12.

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Creíble	100 % (3)	77,70 % (9)	86,67 % (15)
Poco Creíble	85,11 % (47)	75,00 % (12)	75,00 % (8)
No Creíble	100 % (5)	0 % (0)	0 % (0)
Relacionados con información	80,00 % (55)	80,95 %(21)	91,30 % (23)
Relacionados sin información	71,40 % (14)	66,66 % (3)	28,57 % (7)
No relacionados	44,44 % (18)	100 % (2)	100 % (1)
Resto	63,60 % (11)	100 % (1)	0 % (0)

Tabla 4.12: Resultados de precisión (número de muestras) de la clasificación final

Como puede verse en la tabla, el sistema teóricamente debe de funcionar bien, pues los dos niveles de clasificación que más preocupan (nivel C1 y nivel R1) son en los que mejor se están desenvolviendo los clasificadores que se han simulado. Hay que destacar que el sistema con la implementación final del programa *Sniffer* puede variar debido a las librerías que se utilicen. Por lo que se deben tomar los datos procedentes de los resultados con cierto cuidado. Además los resultados se deben de relativizar, las muestras que se han utilizado en ciertos niveles son, hasta cierto punto; insuficientes para determinar el devenir de la clasificación final.

Para concluir, el programa *Sniffer* es un sistema de ayuda que va a tratar de simular escenarios de *crowd-Sensing* para analizar la credibilidad de la información de fuentes sociales basándose en un contexto. Para realizar esta tarea se van a implementar algoritmos de clasificación basados en los algoritmos de aprendizaje automático que se han desarrollado a lo largo de capítulo.

Capítulo 5

Descripción de la Propuesta

A lo largo de este capítulo se va a tratar la estructura y el funcionamiento del programa "Sniffer". Se tratará de presentar la tecnología que se ha usado para realizar el sistema de extracción de tweets, almacenamiento y clasificación así como los algoritmos implementados.

5.1 Sistema de análisis

El sistema de análisis y procesado de datos denominado *"Sniffer"* es la propuesta que se ofrece como sistema de clasificación de la credibilidad de los tweets. Es un software desarrollado sobre Python 2.7[1] y está planteado para aportar una funcionalidad completa al usuario, abarcando desde la recolección de los tweets pasando por su análisis y terminando con la visualización. Todo este sistema está enfocado a ser ejecutado en el lado del servidor y que el cliente acceda a los resultados por medio de navegadores web.

El sistema *Sniffer* presenta diferentes módulos de funcionamiento, la estructura de estos módulos y del sistema en general es la representada en la figura 5.1. La explicación del funcionamiento de los diferentes módulos se realizará a lo largo de los siguientes apartados, excepto el de GraphicInterface que será desarrollado a lo largo del capítulo 6.

[1] http://www.python.org

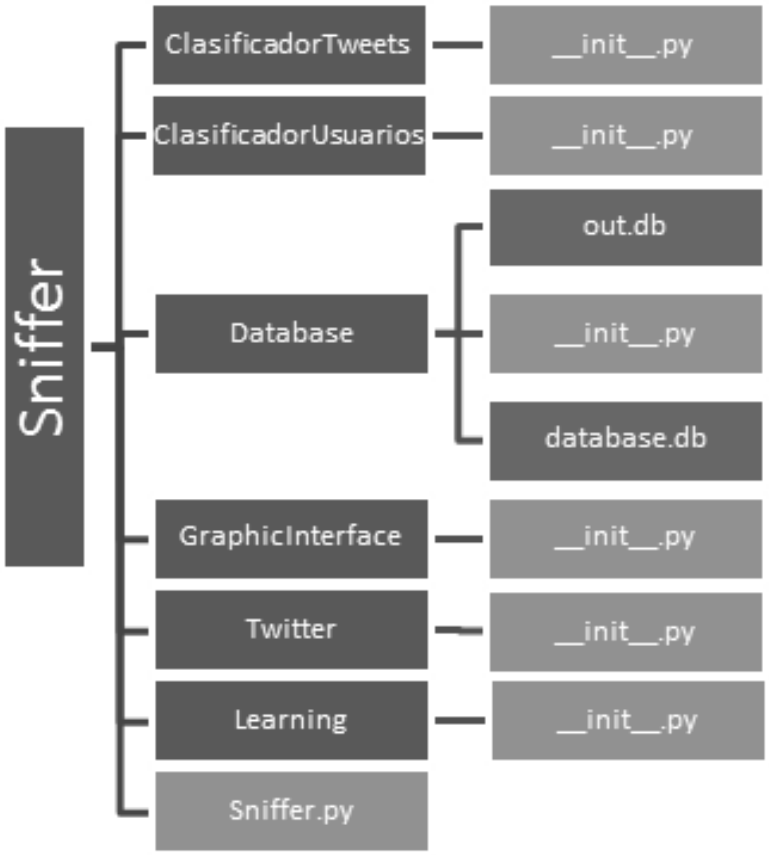

Figura 5.1: Diagrama de Estructura de *Sniffer*

Como se verá en el apéndice F el programa *Sniffer* tiene diferentes modos de operación. De entre estos modos de operación los más relevantes para el proyecto, y los que se van a exponer en este capítulo; son los siguientes:

- **Recogida**: Este modo de operación se encarga de realizar la recolección de los diferentes tweets. Se consulta a Twitter cuáles son los Hashtags disponibles, en nuestro caso en España; y se ofrece al usuario la opción de seleccionar uno de esos contextos o aportar él mismo el que desee. Una vez establecido el Hashtag se empieza a recolectar los tweets.

- **Estudio**: Este modo de operación se encarga de realizar el experimento con los tweets que se han recolectado. A lo largo del apartado 5.4 se exponen los diferentes algoritmos que se han implementado para este proceso.

5.2 Relación con Twitter

Dado que se está trabajando en Twitter y se requiere de la extracción de tweets se debe de implementar algoritmos que permitan su extracción. Cuando se desarrolla el programa sobre Python se tiene diferentes posibilidades para realizar peticiones a los Servicios de Twitter para que aporten información. De entre todas las opciones, las dos más utilizadas son:

- TwitterApi
- Tweepy

De estas dos librerías, la que se va a utilizar va a ser la primera, el motivo de esto es que es más intuitiva con su uso y los resultados que devuelven las peticiones presentan un formato JSON. Los tipos de peticiones que se van a realizar a la plataforma van a ser para los siguientes motivos:

- Extracción de tweets por Hashtag: Este mecanismo permite la extracción de los tweets con la información necesaria para que la aplicación pueda funcionar.
- Extracción de los datos de usuario: Este mecanismo permite la extracción de toda la información relacionada con el usuario que va a ser utilizada por el programa.

De forma previa a la ejecución de estas funciones, se debe de autenticar siempre la aplicación. Para ello se debe de hacer uso del procedimiento OAuth expuesto a lo largo del *Estado del arte* (Capítulo 2).

Cada una de estas fases va a ser desarrollada a lo largo de este apartado, exponiéndose como se extraen y que información nos proporcionan. Todos estos procedimientos están implementados en el módulo *Twitter* a lo largo de la clase *Funciones* (__init__.py).

Dado que Twitter presenta unas fuertes limitaciones en el uso de sus servicios a través de la API, la premisa principal de este módulo debe ser la de minimizar el número de consultas a los Servicios de Twitter.

5.2.1 Extracción de tweets por Hashtag

El sistema *Sniffer* siempre va a realizar la clasificación de los tweets en base a un contexto, es por ello por lo que se necesita que los tweets que se obtienen, pertenezcan a una misma temática. Para establecer esta temática, los Servicios de Twitter nos facilitan el poder realizar la consulta de los tweets en base a una tendencia o *Hashtag*. Para cada uno de estos tweets, los Servicios de Twitter aportan una serie de metadatos asociados a cada uno de los tweets proporcionados en cada petición. De entre todos estos metadatos, para que el programa pueda realizar correctamente su actividad, es necesario que se presenten los elementos dispuestos a lo largo de la tabla 5.1.

Parámetro	**Descripción**
Id_tweet	Identificador único del tweet, el valor de este campo sigue un orden ascendente y lineal, es decir si un tweet posee un id inferior que otro significa que se ha publicado antes.
Num_rt	Número de retweets (redifusiones) que se han aplicado sobre ese tweet.
Fecha	Fecha de publicación del tweet.
Screen_name	Nombre de usuario (@...) en la red social.
Texto	Texto de 140 caracteres el cual contiene la información que se va a transmitir.
Aplicación	Soporte sobre el que se ha publicado el tweet.
Id_user	Identificador único del usuario, el valor de este campo, al igual que con el identificador del tweet, sigue un orden ascendente y lineal.
Followers	Número de seguidores del usuario.
Friends	Número de seguidos del usuario.
Statuses	Número de tweets publicados por el usuario
Localización	Localización del autor del tweet
URL	URL presente en la descripción del usuario.
Coordenadas	Coordenadas del autor en el momento de la publicación del tweet
Nombre	Nombre del usuario en la red
Descripción	Descripción en el perfil del usuario
Idioma	Idioma de la publicación

Tabla 5.1: Parámetros de Funcionamiento

Es a partir de todas estas variables desde las que se obtienen los parámetros necesarios para la clasificación de los tweets que se han desarrollado

a lo largo del capítulo 4: Estudio de los Metadatos. Dicha extracción de parámetros se realiza a lo largo del módulo *ClasificadorTweets*.

Una cosa que particulariza a la red social Twitter es que se pueden realizar redifusiones de los mensajes (*retweets*). Este tipo de tweets ya no son de la autoría de la cuenta en la que se publica, sino que son de otros autores. Estos *retweets* se caracterizan por presentar la siguiente estructura en el texto del tweet:

"RT @Nombre: Texto del tweet"

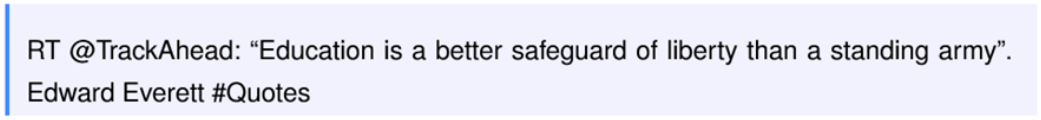

Figura 5.2: Ejemplo de retweet

Estos *retweets* pueden ser útiles, porque pese a no ser tweets propios del autor, el autor original figura en dicho tweet. Es por ello por lo que se pueden modificar los datos que se están registrando para asociarselos al autor original. Este proceso se enmarca dentro del proceso de extracción de los tweets que corresponde con el método *get_trend_Tweet()*. Con la finalidad de poder extraer los datos del usuario original se necesita implementar el procedimiento representado en el diagrama de flujo de la figura 5.3.

Como se puede ver en el diagrama de flujo (figura 5.3) cuando se poseen registros de tweets de la misma tendencia (Hashtag o contexto), se obtiene el último tweet, es decir el de menor id. Esto permite solicitar a Twitter que se proporcionen tweets anteriores a ese. Esto es muy importante porque los Servicios de Twitter no permiten extraer más de 100 tweets por cada petición y a una tasa máxima de 450 tweets por cada 15 minutos (30 tweets por minuto).

Por otro lado cuando se detecta que un retweet, lo que se trata de hacer es obtener el usuario que ha generado el tweet inicial para asociarle a él toda la información. Para ello se comprueba si la primera palabra del tweet es *"RT"* y en caso afirmativo la segunda palabra es el nombre del usuario. Se consulta a la base de datos si se tienen registros de ese usuario y en caso negativo se solicita a Twitter la información necesaria. Una vez obtenidos los datos del usuario, se sustituyen en los registros del tweet correspondientes.

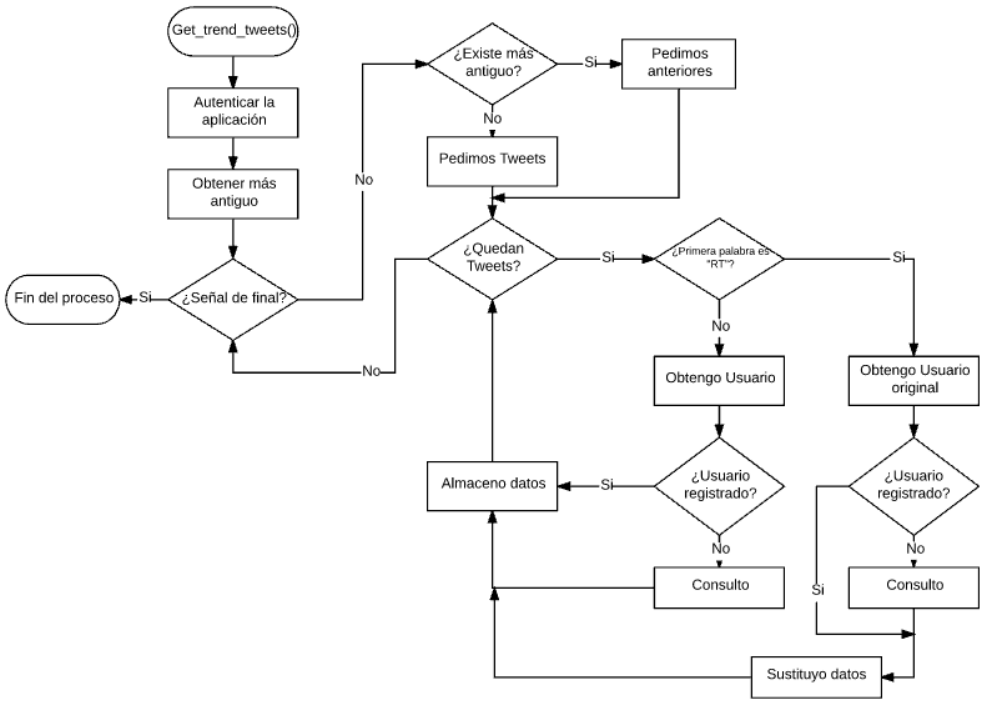

Figura 5.3: Diagrama de flujo de Extracción de tweets

El sistema que se ha implementado, permite un funcionamiento continuo de la aplicación, debido a que se utilizan múltiples permisos de aplicación para acceder a los datos y porque no se superan en ningún momento las tasas máximas permitidas.

5.2.2 Extracción de los Datos del Usuario

Con la finalidad de realizar consultas innecesarias para extraer los datos, cuando se solicita la búsqueda de un usuario es necesario consultar si ya se poseen los datos de usuario para, en caso de que existan, devolverlos y en caso de que no exista solicitarlos a Twitter. La secuencia de ejecución de este procedimiento, registrado en el método *user_analytics()*, es la que se presenta a lo largo del diagrama de flujo de la figura 5.4.

La estructura de los datos que se registran en las bases de datos y todo lo relacionado con la estructura de relaciones entre las diferentes tablas se presenta a lo largo del siguiente apartado.

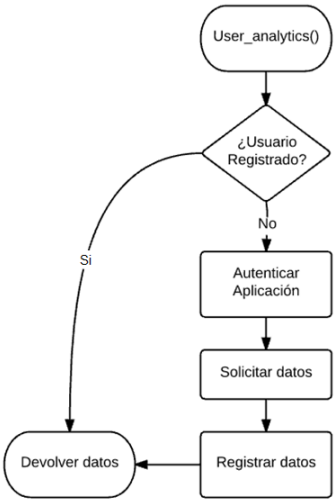

Figura 5.4: Diagrama de flujo de Consulta de Usuarios

5.3 Almacenamiento de Datos

Una parte fundamental del sistema *Sniffer* es el almacenamiento de los datos. El motivo principal de esto, como ya se ha dicho, es que se deben de minimizar el número de consultas realizadas a los Servicios de Twitter. Es por ello por lo que se decide que se debe de implementar una base de datos propia para el funcionamiento del sistema.

Para la creación de la base de datos, se recurre a bases de datos relacionales pues con esto se obtienen las siguientes funcionalidades:

- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

La implementación de la base de datos se va a realizar por medio de SQLite3, se van a realizar 2 bases de datos: *database.db* y *out.db*. Dentro de la base de datos *database.db* se van a establecer 4 tablas diferentes, estas bases de datos tienen el siguiente objetivo:

- **Tweets**: Es donde se almacenan los datos de los tweets que se han extraído. Los parámetros que almacena son los registrados en la tabla 5.1 y adicionalmente se le añade el Hashtag al que pertenece cada tweet.

- **Users**: Es donde se almacenan los datos de los usuarios que se han extraído. Los parámetros que se registran son: identificador del usuario, edad de la cuenta, número de tweets publicados, número de seguidores, número de seguidos, si el usuario esta verificado, si el usuario contiene descripción y si el usuario posee una URL.

- **Train_tweets**: Es donde se registran todos los tweets que han sido clasificados manualmente y que se van a utilizar para el entrenamiento de los clasificadores. Se registran los mismos datos que en la tabla tweets pero adicionalmente se disponen los parámetros *class* y *creíble* con los cuales se determinan las clases a las que pertenecen.

- **Credentials**: Se encarga de almacenar los parámetros necesarios para la autenticación en los Servicios de Twitter. Estos parámetros son los mismos que en la tabla 2.1

Las relaciones entre las diferentes tablas dentro de la base de datos (*database.db*) son las que han sido representadas en la figura 5.5

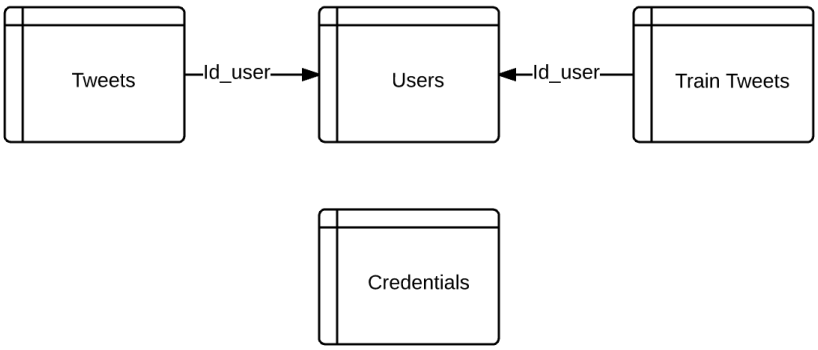

Figura 5.5: Esquema de Relaciones en la Base de Datos

Un aspecto clave en la gestión de la base de datos es minimizar el número de consultas a la misma, es inviable el tener que realizar 2 consultas por cada tweet (Obtención de los datos del tweet y obtención de los datos del usuario). Por ello lo que se trata de realizar es cargar los datos que más frecuentemente se solicitan a la base de datos en memoria, el bloque de tweets relacionados con un Hashtag y los usuarios que los han publicado.

Por otro lado, se ha implementado la base de datos *out.db*. Esta base de datos se encarga de recoger todos los resultados que genera el programa. Consta de una única tabla en la que se almacenan la fecha de creación del

tweet, los identificadores de los tweets, usuario y *screen_name* del usuario además del Hashtag con el que se ha obtenido y de la clase y grado de credibilidad a la que pertenece el registro analizado.

Todas estas implementaciones se están realizando desde el modulo *Database* y en particular desde la clase *DBManager* (__init__.py). Se ha seleccionado SQLite3 como gestor de base de datos, además de porque permite bases de datos relacionales, porque Python lo implementa de forma nativa desde la versión 2.7, por tanto no es necesario de introducir ninguna librería externa y permite almacenar hasta 2 Terabytes de datos.

5.4 Funcionamiento de Algoritmos

Este apartado se va a centrar en exponer la funcionalidad de *Estudio* del programa. La forma en la que está estructurada este modo de operación se puede representar por medio de una máquina de estados como la que se representa en la figura 5.6.

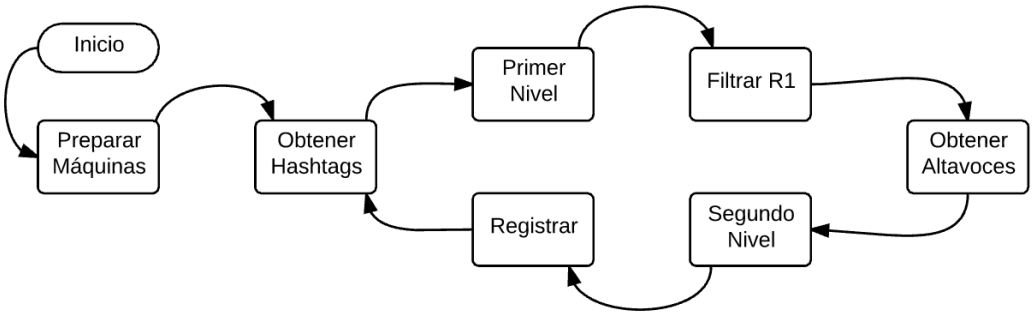

Figura 5.6: Máquina de Estados del modo *Estudio*

Debido a la existencia de ciertos algoritmos que no son triviales de desarrollar, a lo largo de este apartado se trata de exponer los más complejos dentro del sistema *Sniffer*. Estos algoritmos son los que están relacionados con:

- Funcionamiento del programa.
- Secuencia de entrenamiento de los clasificadores.
- Secuencia de obtención de los altavoces de la tendencia.

La estructura global del sistema, junto con las relaciones entre los diferentes elementos de los que consta, es lo que se representa en la figura

5.7 donde las conexiones con las bases de datos se realizan por medio del módulo *Database*. Los bloques que generan una mayor complejidad de implementación son los que se van a tratar de desarrollar en los posteriores apartados con la finalidad de una mejor comprensión del sistema.

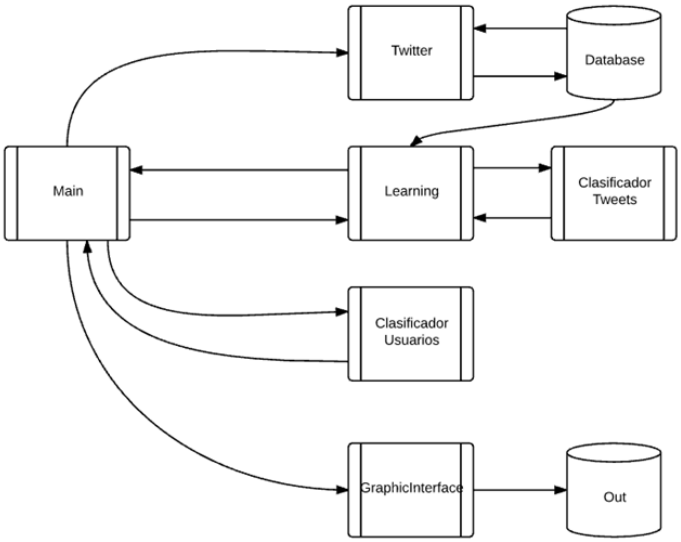

Figura 5.7: Diagrama de Bloques del Sistema *Sniffer*

El programa *Sniffer* presenta dos modos de ejecución, estos modos lo único en lo que se diferencian es en lo que imprimen en la consola durante la ejecución. El modo básico no imprime los resultados del proceso de entrenamiento de los clasificadores, mientras que el modo de prueba imprime en la consola los resultados del proceso de entrenamiento: probabilidad de error y precisión de cada clase.

No obstante todo lo relacionado con el uso del programa *Sniffer* va a ser desarrollado a lo largo del apéndice F (Manual de Usuario).

5.4.1 Algoritmo Principal

El algoritmo principal de funcionamiento (función *Estudio*) se encuentra en el archivo *Sniffer.py*. Este archivo contiene el *main* de la aplicación. La llamada para la ejecución básica del programa se realiza sin introducir ningún tipo de parámetro en la línea de ejecución. Sin embargo, para ejecutar el modo de prueba es necesario introducir en la ejecución un *"-p"*.

Sea cual sea el modo de ejecución, ambos modos lo que realizan es lo representado en el diagrama de flujo de la figura 5.8. Hay que destacar que tanto el entrenamiento de los clasificadores como la clasificación de los datos por parte de los clasificadores automáticos se realizan en 3 bloques independientes. Cada uno de estos bloques se corresponde con cada uno de los grupos en los que se han clasificado los usuarios, esta clasificación es la descrita a lo largo del apartado 4.3.

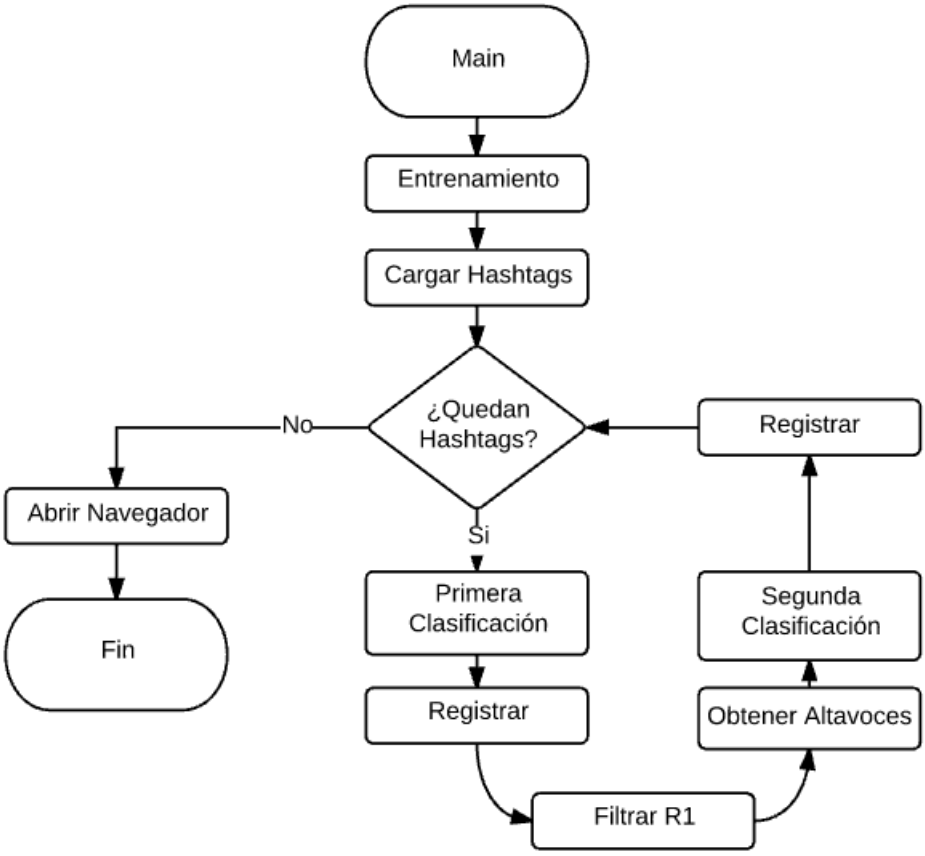

Figura 5.8: Diagrama de flujo Principal del programa *Sniffer*

De los resultados procedentes de la primera clasificación, únicamente interesan los resultados catalogados como *"Relacionado y Contiene Información"* (R1), por ello se guardan en la base de datos de salida todos los datos que no sean R1. Para liberar memoria, se realiza un filtrado de todo el conjunto de datos clasificados y el programa se queda únicamente con R1.

Como se ha comentado, este programa para facilitar la visualización de los datos, clasifica a los usuarios en "*altavoces*". Esta clasificación, que se explica el funcionamiento en el apartado 5.4.3; no influye en el flujo que siguen los tweets, por lo que su siguiente paso es la segunda clasificación y el posterior registro de los resultados de estos datos (hasta ese momento no habían sido registrados en la base de datos de salida los tweets catalogados como R1).

Finalmente, el programa como último proceso realiza la apertura de un navegador en la que se solicita el recurso *inicio*.

5.4.2 Algoritmo de Entrenamiento

Este proceso es la parte fundamental del programa. Para los clasificadores se ha recurrido a los algoritmos de aprendizaje automático expuestos a lo largo del capítulo 4. Para la implementación de estos clasificadores se ha recurrido a la librería *scikit-learn*[21], la cual proporciona una interfaz de trabajo rápida y con muchas posibilidades de ajuste sobre los parámetros de los algoritmos.

Como se ha explicado, existen dos niveles de clasificación aunque ambos actúan de forma semejante. Con la finalidad de mejorar las prestaciones de la clasificación, se realiza una segmentación en el conjunto de todos los usuarios. Esta segmentación se realiza tanto en el proceso de entrenamiento como en el de clasificación y se implementa en el momento de la consulta a la base de datos.

El proceso que siguen los datos a lo largo del entrenamiento de los clasificadores es el que se representa en el diagrama de flujo de la figura 5.9.

El proceso de preparación de las máquinas realiza una partición de los datos en dos conjuntos: *Test* y *Train*. Esta división se realiza en el proceso *Particionar* (en el programa *get_test()*). Este proceso lo que trata es de dividir el conjunto de datos clasificados en dos, intentando que como mínimo exista una muestra dentro del conjunto de entrenamiento (*Train*) y una tercera parte de las muestras que se dediquen a validación (*Test*). Una vez realizada esta partición de los datos, únicamente se entrena con el conjunto *Train* y se valida con el de *Test*.

Dado que se están utilizando múltiples clasificadores de forma simultánea,

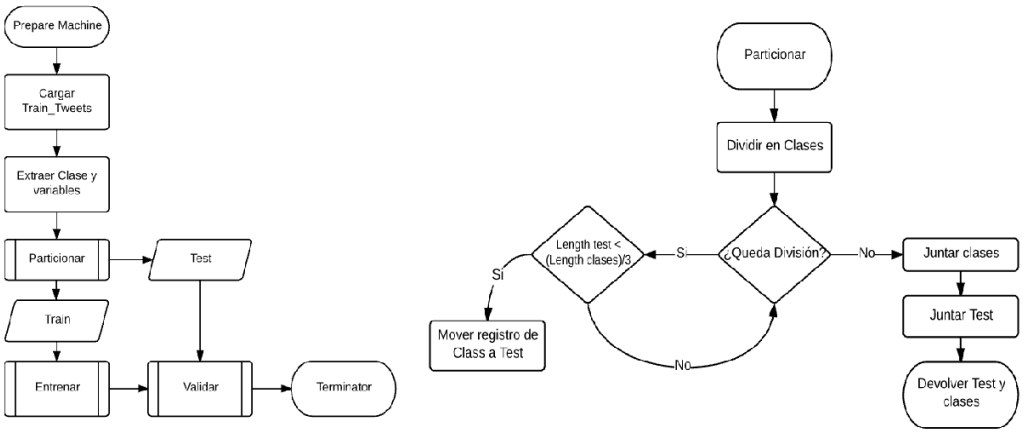

Figura 5.9: Diagrama de flujo del Proceso de Aprendizaje y Validación(Izquierda), proceso de partición del conjunto (Derecha)

es necesario realizar una elección de entre los resultados que proponen. Para ello se implementa la función *map_validator()*, el diagrama de flujo de este proceso de selección está representado en la figura 5.10.

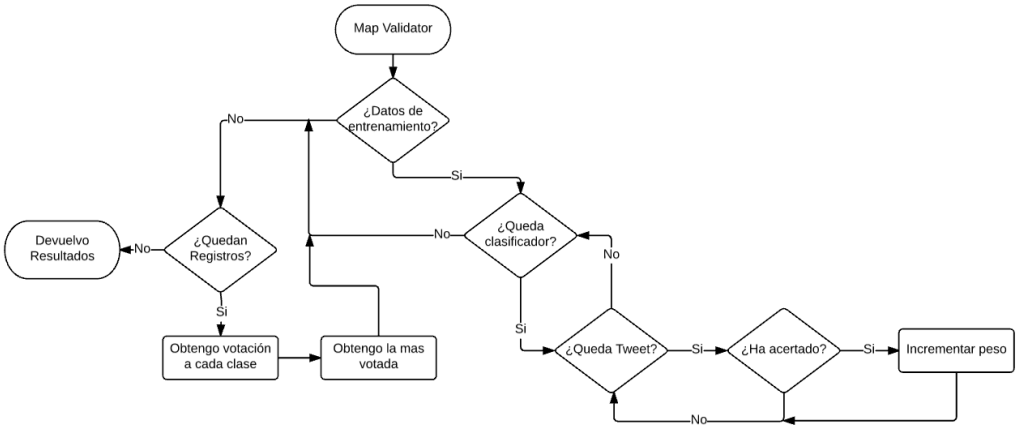

Figura 5.10: Diagrama de flujo de Selección de Clase

Para el proceso de selección se realiza un refuerzo positivo del comportamiento de los clasificadores, es decir, cuando se pasa el conjunto de validación, previamente se comprueba si el resultado arrojado por los clasificadores es correcto; en caso de ser correcto se incrementa el peso que tiene asociado cada clasificador. A la hora de escoger entre el resultado de los clasificadores se realiza la suma de todas las probabilidades de cada clase ponderadas por el peso que se ha determinado. El máximo de estos registros

es el que se toma como verdadero y es el que se le asocia a la muestra. Este proceso es idéntico en los dos niveles de clasificación.

Para cada nivel los clasificadores que se han implementado han sido los siguientes:

- **Primer Nivel**: *Árbol de Decisión*, *Random Forest*, *KNN* y *MLP*
- **Segundo Nivel**: *Árbol de Decisión*, *Random Forest* y *KNN*

Nivel	Clasificador	Variable	Valor
Primer Nivel	**Árbol de Decision**	Criterio	Gini
		Profundidad Máxima	20
	KNN	min_samples_split	4
		min_samples_leaf	3
		K	3
		Métrica	Euclídea
	Random Forest	Algoritmo	Fuerza Bruta
		Pesos	Uniforme
		Profundidad Máxima	20
		Número de estimadores	10
	MLP	min_samples_split	4
		min_samples_leaf	2
		Capas ocultas	(100, 70)
		Activacion	Logística
		Algoritmo	l-bfgs
		Iteraciones	500
		Tasa de aprendizaje	0.1
		Parada	Sí
Segundo Nivel	**Árbol de Decision**	Criterio	Gini
		Profundidad Máxima	20
	KNN	min_samples_split	4
		min_samples_leaf	3
		K	3
		Métrica	Euclídea
	Random Forest	Algoritmo	Fuerza Bruta
		Pesos	Uniforme
		Profundidad Máxima	8
		Número de estimadores	22
		max_features	4

Tabla 5.2: Valores de los Parámetros de los Clasificadores

Hay que denotar, que los clasificadores tienen múltiples parámetros que establecen el devenir de la clasificación, por tanto es indispensable concretar cuáles son los valores de dichos parámetros. Estos valores son los que figuran en la tabla 5.2.

Como puede verse, se ha decidido por implementar 3 clasificadores en el segundo nivel de clasificación. Esto es debido a que las prestaciones que aportaba el algoritmo basado en el *árbol de decisión* no eran suficientes como para que funcionase bien el sistema. Esto se comprobó, como se verá en el capítulo 7; durante el periodo de pruebas.

5.4.3 Algoritmo de Obtención de Altavoces

Este algoritmo, como se ha comentado, no es primordial en el proceso de clasificación de los tweets. Sin embargo, resulta muy útil a la hora de realizar la visualización de los resultados que genera el programa. Este procedimiento lo que trata de realizar, es la búsqueda de los autores que mayor repercusión han tenido dentro del contexto para el que se han generado. Este algoritmo ha sido planteado debido a que muchos tweets son copias de otros y Twitter no se los asocia al autor original del tweet, por tanto el programa *Sniffer* trata de asociar esos tweets a sus verdaderos autores.

Debido a motivos computacionales, la obtención de estos altavoces, se realiza únicamente para el conjunto clasificado como R1 (Con información y relacionado). Este algoritmo está basado en el que plantea la autora de [14] para la plataforma T-Hoarder pero con una serie de modificaciones.

El algoritmo que se plantea en el programa *Sniffer* trata de obtener la similitud entre dos mensajes para, en caso afirmativo; asociarlo al autor más antiguo que haya publicado el tweet y se asocia como si fuese un retweet del original. Para extraer el grado de similitud que presenta el tweet con el original lo que se plantea es utilizar la distancia de *Levenshtein*[22]. Esta distancia mide el grado de similitud que se presentan entre dos cadenas de texto. La ratio que proporciona esta distancia presenta valores de entre 0 y 1, siendo 1 que son iguales y 0 que son completamente distintos. El programa *Sniffer* va a considerar que 2 tweets se pueden considerar copia cuando la ratio sea mayor o igual que 0,9. No se utiliza 1 debido a que muchos tweets pueden tener menos o más caracteres y estar expresando lo mismo, por ello se deja ese cierto margen.

Todo este procedimiento esta implementado dentro del módulo de *ClasificadorUsuarios* en el fichero __init__.py. El algoritmo que se implementa para desarrollar este proceso es el que se representa a lo largo del diagrama de flujo de la figura 5.11.

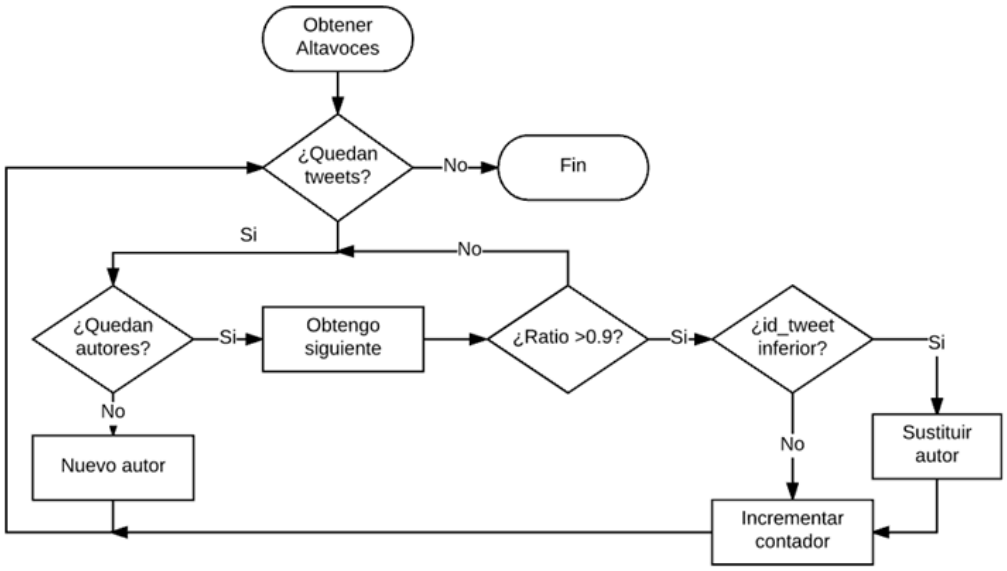

Figura 5.11: Diagrama de flujo de Obtención de Altavoces

Como puede verse en el diagrama de flujo esta tarea requiere de un coste computacional muy alto debido a que se debe comparar cada tweet nuevo con todos y cada uno de los tweets que se tienen registrados, por ello se hace necesario la implementación de ciertos algoritmos para aliviar el número de autores con los que se deben de comparar. Para esto se establece un truncamiento en el conjunto de los autores. Este truncamiento se realiza en cada vez que se analizan 5000 tweets. Cuando se analiza esta cifra de tweets el truncamiento aplicado puede ser de 2 tipos:

- Si el número de autores es superior a 1000, se eliminan a todos los autores que tengan un número de retweets asociados inferior a la mitad de la media de retweets entre todos los autores.
- Si el número de autores es inferior a 1000, se eliminan todos aquellos autores que tengan un número de retweets inferior o igual a 1.

Cuando se aplica este procedimiento la representación no varía y además se consigue que el programa tarde menos y ocupe un espacio un menor en memoria. Esto se debe a que se pasa de tener más de 25000 autores (para 67000 tweets) a tener apenas 3000 autores.

Finalmente se obtienen los autores *Top*, creándose dos clasificaciones de usuarios en función del número de retweets totales que han obtenido en el contexto. Estos dos top son los siguientes:

- **Top20**: A este nivel únicamente acceden aquellos autores que reúnen entre todos más del 20 % de los retweets del contexto.

- **Top50**: A este nivel únicamente acceden aquellos autores que reúnen entre todos más del 50 % de los retweets del contexto y que no están en el Top20.

Una vez obtenidos todos los autores el proceso que se realiza es la clasificación de los altavoces. Para asociar una clase u otra se necesita del uso de dos parámetros adicionales:

- **k_in**: Este parámetro me determina el número de retweets en media que han tenido sus tweets publicados en el contexto, es decir (número de retweets)/(número de tweets).

- **ratio**: Este parámetro es el que desciende de la clasificación de los usuarios (Capítulo 4.3) y es la ratio entre seguidores y seguidos del usuario.

Con los tops y con estos parámetros se realiza la clasificación de los usuarios, éstos se pueden clasificar en cuatro tipos de altavoces:

- **Altavoz Alto**: Estos autores son los que más repercusión han tenido en el contexto. Se caracterizan por tener un *k_in* superior a 3, un *ratio* mayor que 10 y estar presentes en el *Top20* del contexto.

- **Altavoz Medio**: Estos autores son los que han tenido repercusión en el contexto, pero menos que los altavoces altos. Se caracterizan por tener un *k_in* superior a 3, un *ratio* mayor que 10 y estar presentes en el *Top50* del contexto.

- **Altavoz Bajo**: Estos autores son los que han tenido poca repercusión en el contexto, menos que los altavoces medios. Se caracterizan por tener un *k_in* superior a 3, un *ratio* mayor que 10 y no estar presentes en ninguno de los *top*.

- **No Altavoz**: Estos autores son los que menos repercusión han tenido dentro del contexto. Se caracterizan por, tener un *k_in* inferior a 3 o una *ratio* inferior a 10, es decir ser un Usuario Normal.

Con esto ya se tienen diferenciados a los usuarios en función de cómo se desenvuelven dentro del contexto del que se han extraído los tweets.

Capítulo 6

Visualización de los resultados

A lo largo de este capítulo se va a desarrollar el sistema que se ha implementado para la visualización de los resultados procedentes del programa Sniffer. La representación de estos resultados es peculiar debido a que se están tratando con datos personales. Estas peculiaridades son las que se resolverán a lo largo de este capítulo.

6.1 Introducción

Con la finalidad de poder comprender de forma más sencilla los resultados que se arrojan en el programa *Sniffer*, se pretende realizar una representación de los resultados. Para poder desarrollar este sistema de visualización, se ha requerido de la utilización de servidores web y una base de datos donde registrar todos los resultados del programa. Esta base de datos ya se ha mencionado a lo largo del presente documento siendo la base de datos *out.db*

La representación de los resultados procedentes de la ejecución del programa *Sniffer* se van a realizar de forma separada en función del contexto al que pertenezcan los registros. De cada uno de los contextos registrados se van a elaborar dos tipos de diagramas:

- **Diagrama Circular**: Este diagrama pretende representar la distribución de las diferentes clases de tweets que han sido analizados.
- **Diagrama Temporal**: Este diagrama pretende representar la evolución temporal que ha sufrido cada clase durante la *"vida"* del contexto.

No obstante, a lo largo de este capítulo se va a plantear el desarrollo del sistema de visualización en dos partes:

- **Elaboración de resultados**
- **Implementación del servidor**

6.2 Elaboración de resultados

La implementación del sistema que se encarga de realizar el registro de los resultados se lleva a cabo dentro del programa *Sniffer* en el módulo *GraphicInterface*.

Este módulo se encarga de registrar todos los resultados dentro de la base de datos *out.db*. Los datos que se registran dentro de esta base son:

- **Fecha de creación**: Determina la fecha de creación del tweet al que está asociada.
- **Identificador del tweet**: Es el identificador único e inmutable que se asocia al tweet.
- **Identificador del usuario**: Es el identificador único e inmutable que se asocia al usuario autor del tweet.
- **Texto**: Es el contenido que se está analizando dentro del tweet.
- **Clase**: Representa la clase que se ha asociado al contenido en el primer nivel de clasificación.
- **Creíble**: Representa la clase que se ha asociado al contenido en el segundo nivel de clasificación (si procediese).
- **Contexto**: Representa el contexto al que esta asociado el registro.

Sin embargo existe una peculiaridad dentro del atributo *Clase*. Debido a que se realiza una diferenciación dentro antes del segundo nivel de clasificación en función de la repercusión que ha tenido el tweet en el contexto, se utiliza este atributo para identificar la clase de *"altavoz"* que se le ha asociado al usuario autor del tweet.

Es por ello por lo que cuando un tweet ha accedido al segundo nivel de clasificación este parámetro cambia su significado pasando a ser *5*, *6*, *7* y *8*

cuando los usuarios son catalogados como *"altavoz alto"*, *"medio"*, *"bajo"* y *"no altavoz"* respectivamente.

El registro de cada uno de estos datos se realiza una vez el tweet ha finalizado el proceso de clasificación. Esto se realiza con la finalidad de liberar la memoria lo más rápido posible para agilizar el proceso de clasificación.

6.3 Implementación del servidor

Como se ha comentado, la representación de los resultados que genera el programa se va a realizar en un servidor web. De entre todas las opciones que hay, para crear esta estructura se va a utilizar un servidor Apache[1]. La versión de servidor que se está utilizando es *apache-tomcat-8.0.28*.

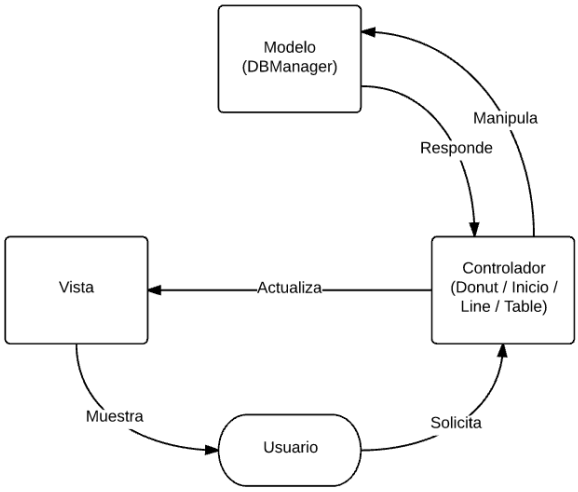

Figura 6.1: Patrón Modelo-Vista-Controlador

El modelo que se ha utilizado para la implementación del servidor ha sido el patrón *Modelo-Vista-Controlador* donde se ha dispuesto una serie de controladores para capturar cada una de las posibles peticiones a los recursos que posee el sistema. Además, para poder transferir más fácilmente la información a representar, se ha decidido implementar una serie de controladores para transmitir los contenidos en tiempo real. Para ello se ha utilizado la herramienta AJAX.

[1]https://httpd.apache.org/

Dentro de este sistema de clasificación se presentan diferentes tipos de vistas. A grandes rasgos se pueden separar en las que representan información propia del contexto y las que representan información genérica de la aplicación. Por tanto la estructura de relaciones entre las diferentes vistas del sistema de visualización es la que se representa en la figura 6.2.

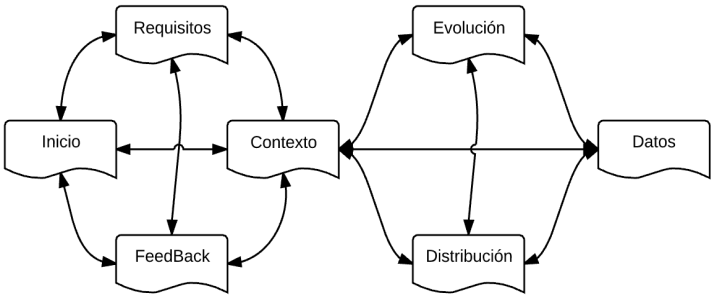

Figura 6.2: Estructura de vistas web

Como puede verse en la figura 6.2, existe una vista llamada *Contexto*, esta vista es la que representa el contenido asociado a cada contexto. El contexto que se debe de mostrar se registra en la sesión del usuario y se actualiza cada vez que el usuario cambia de contexto. En caso de que el usuario intentase acceder a cualquier vista particular sin haber pasado por la vista *Contexto*, se le redirecciona automáticamente a la vista de inicio. Esto es porque el controlador que gestiona la vista *Contexto* es el que se encarga de registrar el contexto la sesión del usuario.

Para cada una de las diferentes vistas asociadas a los contextos, se representan los siguientes elementos:

- **Contexto**: Esta vista pretende mostrar un pequeño resumen de lo que va a poder observar el usuario en el resto de las vistas asociadas al contexto.

- **Distribución**: Esta vista muestra el conjunto de diagramas circulares mencionados anteriormente. En esta vista se representan exactamente cinco diagramas circulares los cuales están asociados a *"primer nivel de clasificación"*, *"altavoces altos"* , *"altavoces medios"* , *"altavoces bajos"* y *"no altavoces"*.

- **Evolución**: Esta vista muestra el conjunto de diagramas temporales mencionados anteriormente. En esta vista se representan exactamente

cinco diagramas temporales los cuales están asociados a *"primer nivel de clasificación"*, *"altavoces altos"* , *"altavoces medios"* , *"altavoces bajos"* y *"no altavoces"*.

- **Datos**: Esta vista muestra los resultados que ha arrojado del sistema con cada uno de los tweets del contexto. En esta vista se representan exactamente cinco tablas de datos las cuales están asociadas a *"primer nivel de clasificación"*, *"altavoces altos"* , *"altavoces medios"* , *"altavoces bajos"* y *"no altavoces"*.

La mayoría de los datos que se representan se pueden obtener directamente realizando la consulta apropiada a la base de datos *out.db*. Sin embargo, el algoritmo para establecer el diagrama temporal presenta una mayor complejidad. Este algoritmo se expone a lo largo de la figura 6.3.

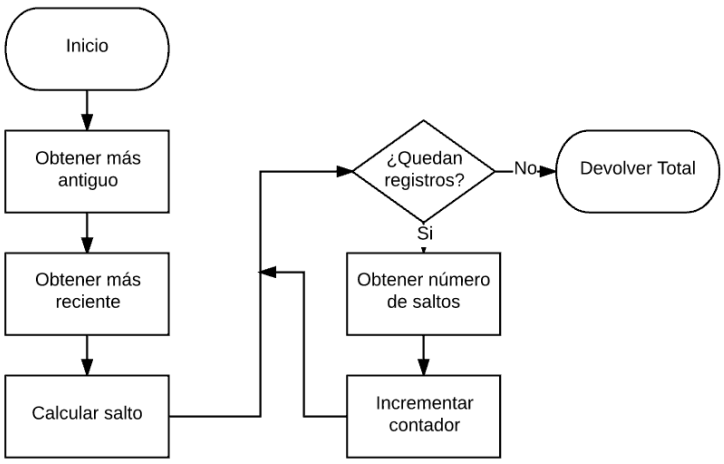

Figura 6.3: Obtención de diagrama temporal

Este algoritmo lo que pretende realizar es una lista de 80 elementos que están compuestos por: tiempo del intervalo al que están asociados y número de elementos en ese intervalo. El intervalo se obtiene a partir del tiempo de duración del contexto. Se decide establecer una serie de 80 intervalos en cada contexto, para ello se divide el tiempo total de vida del contexto entre el número de intervalos requeridos.

Dentro de la lista, la posición que ocupa en el diagrama está determinada por la posición dentro de la lista y por tanto por el intervalo al que pertenece. Finalmente una vez se ha obtenido toda la lista se decide de transmitirla al cliente por medio de un formato JSON donde se transmite cada una de las listas asociadas a cada una de las líneas que se quiere representar en el

diagrama.

Por otro lado, debido a que se está trabajando con datos de índole personal, se debe de ajustar la representación de los mismos a la legislación, en este caso, española. Esta legislación es la expuesta a lo largo del capítulo 3. Es por ello por lo que es necesario aplicar una transformación irreversible a los datos que nos vemos obligados a ocultar. Estos datos son exactamente:

- **id_tweet**
- **id_user**

Es obligatorio ocultar esta información porque son elementos únicos e inmutables dentro de la red social Twitter. Por tanto, se puede vincular directamente estos parámetros con la persona física que los ha generado.

Es por estos motivos por lo que se plantea la aplicación a estos parámetros de un algoritmo de *hash*. Sin embargo estas funciones presentan una vulnerabilidad, es posible diseñar ataques para extraer el elemento previo a la transformación. Es por esto por lo que se decide que se va a aplicar, de forma previa a la función hash, una funcion *XOR* con un número aleatorio generado de forma segura (*Salt*). De esta manera se está protegiendo al objeto representado de esta vulnerabilidad.

Por otro lado se debe de proteger la base de datos contra diferentes tipos de ataques como la *inyección de SQL*, para ello se va a recurrir siempre que se realice una consulta a la base de datos a los *PreparedStatement*. Con este sistema, se está evitando que lo que introduzca el usuario en la consulta pueda modificar o alterar la estructura de la misma, manteniendo seguros los datos registrados.

A lo largo del apéndice F se expondrá como manejar todas estas vistas por parte del usuario.

Capítulo 7

Pruebas

A lo largo de este capítulo se va a desarrollar el programa de pruebas del sistema Sniffer. Para ello se van a estructurar las pruebas en cada uno de los niveles de clasificación del sistema y en la implementación final de la clasificación. Para ello se hará uso del modo "Pruebas"del programa Sniffer

7.1 Introducción

A lo largo de los capítulos 5 y 6, se ha expuesto la información más importante para la comprensión del sistema *Sniffer* que se ha propuesto. Como se ha podido comprender, este sistema está compuesto por dos secciones claramente diferenciadas: el sistema de clasificación y el sistema de visualización.

Es por este motivo, por el que se van a realizar diferentes tipos de pruebas al sistema. Estas pruebas se pueden clasificar en dos tipos:

- Destinadas a probar los clasificadores.
- Destinadas a probar el sistema *Sniffer*.

A lo largo de este capítulo, se trata de exponer cuáles han sido las pruebas realizadas y los diferentes resultados que han aportado.

7.2 Sistema de clasificación

El sistema de clasificación del programa *Sniffer* establece una serie de clasificadores los cuales permiten extraer la credibilidad de los contenidos de los diferentes tweets que se alojan en el contexto para el que han sido publicados.

Esta clasificación, como se ha visto, se realiza en dos niveles de los cuales el primero pretende determinar si un contenido contiene información relacionada o no relacionada y el segundo pretende determinar el grado de credibilidad que se aporta al mensaje transmitido.

La librería Scikit-Learn, permite la posibilidad de extraer una serie de parámetros de cada nivel de clasificación, midiéndose aspectos tales como:

- **Precisión**: Esta medida permite extraer el grado de dispersión del conjunto de valores obtenidos tras realizar diferentes mediciones. Esta relación determina que, cuanto menor es la dispersión mayor es la precisión, es decir, a mayor precisión el número de falsos positivos que van a aparecer en la clasificación será menor.

- **Sensibilidad o *recall***: Esta medida indica la capacidad que tiene el estimador para determinar como casos positivos los casos negativos. De este modo, una sensibilidad mayor permite obtener una menor cantidad de falsos positivos y viceversa. La sensibilidad suele estar ligada a la precisión, debido a que el hecho de mejorar la sensibilidad a menudo conlleva un descenso de la precisión del sistema pues resulta más complejo ser preciso si el espacio entre muestras se incrementa.

- **Medida-F o *F-measure***: Esta medida procede de la combinación de las medidas de precisión y sensibilidad. Su valor no aporta mucho significado en ciertos tipos de clasificadores como los clasificadores bayesianos.

Estas pruebas están destinadas a encontrar las prestaciones que es capaz de aportar el sistema de clasificación perteneciente al programa *Sniffer* en la clasificación de los diferentes tweets del contexto correspondiente. Para ello se va a establecer que las prestaciones que deben de arrojar los diferentes niveles de clasificación deben de ser parejas a los resultados teóricos establecidos a lo largo del capítulo 4.

Sin embargo, es necesario destacar que estos niveles de exigencia pueden verse alterados. El principal motivo de esto es que no todos los usuarios tienen

la misma repercusión dentro de la red social, de aquí que se haya hecho la separación de los usuarios mencionada en el capítulo 4. Con ello se establece una serie de criterios:

- **Usuarios Normales**: Dado que el alcance de estos usuarios suele estar muy limitado, es posible reducir el nivel de exigencia dentro de esta clasificación.
- **Usuarios Corporativos**: El alcance de estos usuarios comienza a ser superior, por tanto se ve necesario incrementar los requisitos de funcionamiento de estos clasificadores.
- **Usuarios Organizativos**: El alcance de estos usuarios es muy elevado, por ello se ve necesario poner unos criterios de aceptación elevados.

Para ello se van a plantear las diferentes pruebas para obtener las prestaciones de cada nivel de clasificación. Para la evaluación del sistema se ha establecido un conjunto de entrenamiento de más de 1700 tweets clasificados de forma manual, estos tweets han sido con los que se ha realizado el estudio de los metadatos del capítulo 4 y son con los que se van a validar los niveles de clasificación del sistema *Sniffer*.

Es necesario denotar que aportar el grado de credibilidad a un contenido puede tener diferentes consecuencias, un ejemplo claro es aportar un grado de credibilidad alto a una información falsa en periodos de emergencia social. Es por esto por lo que en el desarrollo de este sistema se ha mantenido la premisa de que se prefiere aportar el grado de credibilidad inferior a los posibles ante situaciones de duda, es decir, un tweet que sea C1 se prefiere que se clasifique como C2 antes que un tweet C3 se clasifique como C1.

7.2.1 Primer nivel de clasificación

Como se ha tratado de exponer a lo largo de los diferentes capítulos del presente documento, se plantea como objetivo otorgar a este nivel de clasificación la capacidad de discernir en torno a la información que incluye el contenido. El conjunto de datos que se introduce en este nivel de clasificación está segmentado en función del ratio comentado a lo largo del capítulo 4.

Adicionalmente a esta segmentación del conjunto, para este nivel de clasificación se ha realizado una separación en cada uno de los conjuntos de entrenamiento de los datos destinados a la validación del sistema

(aproximadamente el 30 % de cada conjunto) y con el resto ha sido con lo que se ha entrenado a los diferentes clasificadores de este primer nivel de clasificación (*KNN*, *Árbol de decisión*, *Random Forest* y *MLP*).

Con este planteamiento los resultados que se han obtenido en la ejecución de esta prueba han sido los que están expresados a lo largo de la tabla 7.1.

Clase	**Usuarios Normales**	**Usuarios Corporativos**	**Usuarios Organizativos**
Relacionados con información	86 % (100)	93 %(151)	96 % (399)
Relacionados sin información	69 % (34)	92 % (40)	95 % (48)
No relacionados	86 % (26)	92 % (14)	100 % (14)
Resto	62 % (12)	0 % (3)	100 % (6)
Probabilidad de Acierto Media	81,39 %	92,78 %	96,35 %

Tabla 7.1: Resultados de precisión (número de muestras) de la implementación de la primera clasificación

Como puede verse en la tabla 7.1 se está obteniendo unas prestaciones superiores a los niveles de clasificación previstos por medio de las simulaciones, por ello se pueden aceptar las implementaciones que se han realizado de los algoritmos, así como del decisor MAP implementado.

Estas prestaciones se pueden considerar aceptables pues, en los usuarios con un mayor poder de difusión como son los usuarios corporativos y organizativos, se presentan las mejores prestaciones por lo que se puede considerar como que el sistema se va a comportar correctamente en este nivel en cuanto a probabilidad de error se refiere.

Sin embargo, basar el criterio de validación utilizando únicamente la probabilidad de error o en la precisión no determina el correcto funcionamiento del sistema. Para ello se realiza un estudio de la sensibilidad (o *recall*) de los diferentes clasificadores para las diferentes clases de este primer nivel. Los resultados son los que se pueden ver a lo largo de la figura 7.1. Esta medida, como se ha dicho anteriormente, evita la aparición de una gran cantidad de falsos positivos, por tanto es importante tener una sensibilidad alta en la clase *Relacionada y con información* (*R1*) en el primer nivel de clasificación.

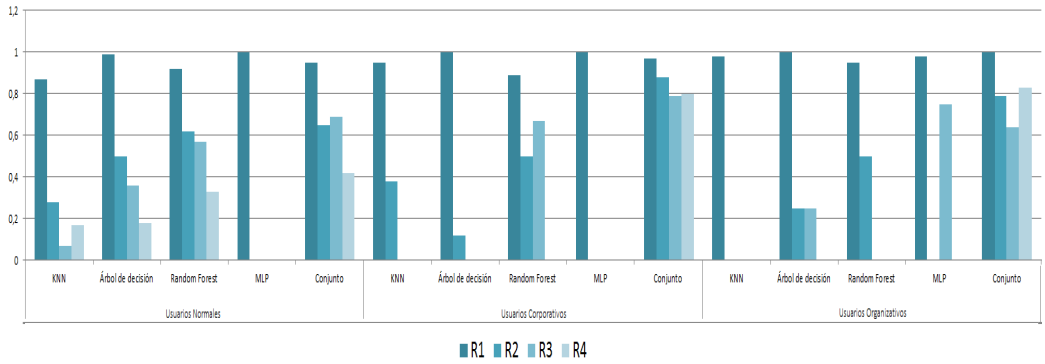

Figura 7.1: Sensibilidad del primer nivel

Como puede verse en la figura 7.1 los diferentes clasificadores proporcionan una sensibilidad en la clase *R1* superior en todos los casos a 0.9 por lo que, en conjunto con la probabilidad de error expuesta anteriormente, se puede deducir que este primer nivel va a funcionar de forma aceptable.

7.2.2 Segundo nivel de clasificación

Para el segundo nivel de clasificación se realiza un cambio en el objetivo de la clasificación, en este nivel se trata de obtener el verdadero sentido del tweet que se está analizando para poder discernir el grado de credibilidad que se le va a aportar. Al igual que en primer nivel, en éste también existe una segmentación de los tweets en función de los usuarios.

Esta prueba se realiza partiendo el conjunto de datos en 2 subconjuntos: *entrenamiento* (70º %) y *validación* (30 %). El conjunto de entrenamiento se va a utilizar para preparar los diferentes clasificadores mientras que el de validación se encargará de probar las prestaciones que ofrece cada clasificador.

Como se puede ver en el sistema planteado en el capítulo 4, para el segundo nivel de clasificación, el sistema que se plantea es el basado en un único árbol de decisión. Sin embargo, a la hora de implementar este único clasificador se obtienen unas prestaciones muy pobres (una probabilidad de acierto cercana al 60 %). Por tanto se ha decidido plantear de nuevo el implementar un sistema de clasificación basado en múltiples clasificadores. Probando con los diferentes clasificadores, se obtiene que los algoritmos basados en *árboles de decisión*, *KNN* y *Random Forest* cuando trabajan en conjunto aportan las prestaciones representadas en la tabla 7.2.

Se han obtenido unas prestaciones muy superiores a las que se han

Clase	Usuarios Normales	Usuarios Corporativos	Usuarios Organizativos
Creíble	71 % (9)	88 % (80)	99 % (310)
Poco Creíble	84 % (45)	84 % (51)	88 % (86)
No Creíble	44 % (11)	100 % (19)	100 % (2)
Probabilidad de Acierto Media	76,92 %	88,0 %	96,23 %

Tabla 7.2: Resultados de precisión (número de muestras) de la implementación de la segunda clasificación

simulado a lo largo del capítulo 4. Con estos resultados es posible ver que el sistema está funcionando muy bien en los entornos en los que necesita la máxima precisión en la clasificación (Usuarios corporativos y usuarios organizativos). Por tanto, a nivel de probabilidad de error, se pueden dar como válidos estos resultados debido a que sea cual sea el nivel de clasificación, se tiene que la probabilidad de acierto es más del doble que si se escogiese al azar la clase a la que pertenecen.

Sin embargo, en este nivel cobra una vital importancia la sensibilidad, pues se está clasificando finalmente la credibilidad del tweet y, por tanto, no se puede otorgar *C1* (*creíble*) a un tweet *C3* (*no creíble*). Por esto mismo, al igual que en el primer nivel de clasificación se analiza cual es el rendimiento en términos de sensibilidad. Este rendimiento es el que se plasma en la figura 7.2.

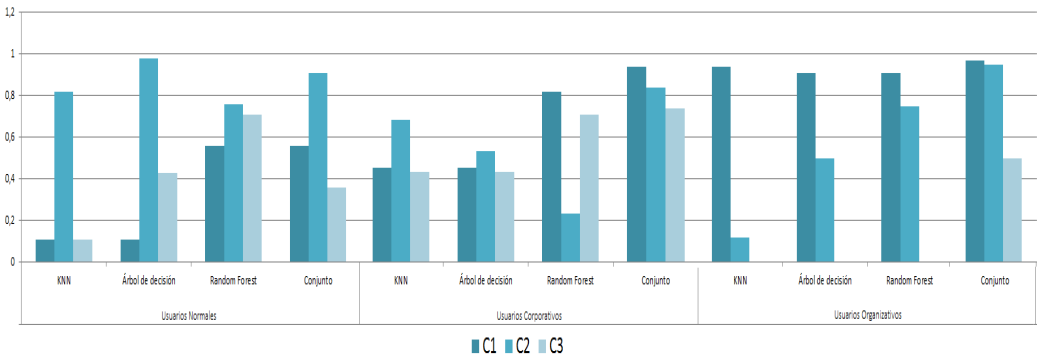

Figura 7.2: Sensibilidad del segundo nivel

A la vista de los resultados se pueden extraer diferentes conclusiones:

- La clasificación en los usuarios organizativos se puede considerar que está funcionando correctamente, debido a que la el grado de máxima credibilidad (*C1*) presenta una sensibilidad superior al 90 % y el de credibilidad media (*C2*) presenta una sensibilidad superior al 60 % por lo que de producirse un falso positivo, tiene una mayor probabilidad de darse en el grado menor de credibilidad (*C3*).

- La clasificación en los usuarios corporativos se puede considerar también válida pues la sensibilidad de los dos niveles de credibilidad más extremos se sitúa por encima del 65 % determinándose que existe una baja proporción de falsos positivos de *C1* procedentes de la clase *C3*.

- La clasificación en los usuarios normales presenta unas prestaciones inferiores al resto de grupos, sin embargo, se puede ver como la sensibilidad de las clases *C2* y *C3* es bastante alta por lo que se permite tener, en cierto grado, acotada la clasificación de estas clases. Es necesario ver que en este nivel la repercusión que se tiene es baja, estos usuarios en media tienen 1018 seguidores (*followers*), por lo que un fallo en este conjunto está más contenido dentro de la red social.

Por tanto, se puede considerar que este sistema está funcionando correctamente en el segundo nivel de clasificación. Estas pruebas no han arrojado información que difiera con lo inicialmente planeado, pues obtener unas prestaciones inferiores en el nivel de usuarios normales es totalmente comprensible pues existen una mayor cantidad de usuarios y, consecuentemente, una mayor variedad de perfiles de tweets.

7.2.3 Sistema completo de clasificación

Para la validación del sistema final se ha procedido a la extracción de tweets pertenecientes a diferentes contextos. De entre los contextos más importantes se presentan:

- **#Bruselas**: Contexto perteneciente a los atentados de Bruselas ocurridos en el mes de marzo de 2016.
- **#SesionDeInvestidura**: Contexto perteneciente a las dos sesiones de investidura fallidas en el año 2016.
- **#Seseña**: Contexto perteneciente al incendio del depósito de neumáticos ocurrido en mayo de 2016 en la localidad de Seseña (Madrid).
- **#Orlando**: Contexto perteneciente al ataque a una sala de fiestas ocurrido en junio de 2016 en la localidad de Orlando (Estados Unidos).

Para poder considerar válidos los resultados, dentro de los diferentes conjuntos de clasificación se han establecido una serie de tweets como referencia, usuarios como *@policia* (Cuerpo Nacional de Policía) o *@MAECgob* (Ministerio de Asuntos Exteriores) son dos claros ejemplos en los que se deben clasificar como C1 en el dataset relacionado con los atentados de Bruselas. Por otro lado, para clasificar tweets que no están relacionados, se poseen tweets relacionados con el fraude de Vitaldent en el cual existen tweets de mofa o chistes sobre el caso. Estos son ejemplos de cómo se ha tratado de comprobar si el sistema funciona bien.

Los resultados arrojados por el programa para estos tweets son muy satisfactorios pues los clasifica siempre de forma correcta. Sin embargo, en caso de que no se consiguiese clasificar un tweet correctamente en el programa, es posible corregir el comportamiento introduciendo dicho tweet manualmente en la base de datos de entrenamiento para futuros experimentos.

7.3 Pruebas de funcionamiento

Las pruebas destinadas a comprobar el funcionamiento del sistema *Sniffer* se pueden dividir en las destinadas a la parte de visualización y en las destinadas a la parte de generación de los resultados.

El objetivo principal del programa es clasificar la credibilidad de los tweets acorde a lo explicado a lo largo del presente documento. El rendimiento de la aplicación en esta parte del proyecto se puede considerar

satisfactoria de acuerdo a lo expuesto en el apartado anterior. Sin embargo, se han realizado diferentes pruebas para comprobar cómo se desenvuelve el programa con diferentes conjuntos de datos atendiendo al tamaño de los mismos.

Para ello los conjuntos de tweets que se han mencionado en la validación del sistema final, se han utilizado también para evaluar el tiempo que tardan en analizarse. Estos conjuntos de datos suman más de 120.000 tweets en total, siendo distribuidos en 67.000 tweets para el hashtag *#Bruselas*, 37.000 para el contexto *#Seseña* y el resto para el contexto *#SesionDeInvestidura*. El tiempo de ejecución de estos tres conjuntos asciende hasta los 50 minutos.

Gran parte de este tiempo es lo que se consume en el proceso de obtención de los altavoces del programa. Para tratar de minimizar este tiempo se ha decidido implementar todos los conjuntos de datos a analizar por medio de diccionarios en lugar de listas. El motivo de esto es que las listas tienen un tiempo de acceso exponencial conforme al tamaño de las mismas y los diccionarios presentan una respuesta lineal conforme al tamaño de los mismos.

Por último, una parte importante del proyecto, como se ha tratado de exponer, es la visualización de los resultados. Para la validación de la fluidez en la navegación a través de la aplicación web se utilizaron estos mismos contextos. Inicialmente en el apartado de *Datos* se cargaban todos los registros a la vez, sin embargo esto es altamente costoso en términos computacionales para el navegador. Por este motivo se decide plantear que en esta vista las peticiones al servidor para cada tipo de dato representado se realicen al solicitar ese conjunto de datos únicamente. Con esto se está mejorando la fluidez pues se reducen los tiempos de carga de las diferentes vistas de la aplicación web.

Capítulo 8

Gestión del proyecto

A lo largo de este capítulo se va a explicar las diferentes fases por las que ha transcurrido el proyecto. Se tratarán las decisiones que se han rechazado a lo largo del desarrollo y las que se han tomado como correctas.

8.1 Evolución del Planteamiento

Conforme se ha ido desarrollando el programa *Sniffer*, se han planteado diferentes puntos de vista para la resolución del problema. Estas fases se pueden estructurar en las siguientes:

- **Análisis de influencias**
- **Análisis de textos**
- **Análisis de metadatos**

A lo largo de este apartado se van a exponer los motivos por los que se han rechazado o se han aceptado cada una de las diferentes opciones.

8.1.1 Relación de influencias

Inicialmente, el proceso de clasificación de la credibilidad del contenido de los tweets se basaba en la calidad de las fuentes de información. Twitter proporciona información en torno a los usuarios y entre esta información es posible extraer si el usuario estaba verificado (existe una persona física real detrás de la cuenta) o no. Con esto se está consiguiendo extraer si la información que recibe un usuario por parte de los usuarios a los que sigue, tiene una base razonada y oficial o no y, por tanto, se le estaba aportando

credibilidad al contenido.

Este punto de vista pronto se puede despreciar debido a que no todos los usuarios verificados de Twitter son expertos en todos los temas, por tanto, unos usuarios pueden producir tweets con un grado de credibilidad superior en unos temas mientras que en otros pueden no ser expertos y transmitir información con un grado de credibilidad muy bajo.

Es este el principal motivo por el que esta opción, por sí sola, se decide rechazar.

8.1.2 Análisis de Textos

Sobre la base de establecer la clasificación de la credibilidad en las relaciones de influencias que existen entre los seguidos de un usuario, se plantea la solución de aportar a los usuarios las clases en las que se les puede considerar expertos. Para ello se realiza un análisis de los 50 últimos tweets que ha publicado cada usuario seguido del usuario que se está analizando y se obtienen los temas que expresan esos tweets.

Con este análisis de los 50 últimos tweets se consigue determinar las clases en las que un usuario se desenvuelve mejor y con ello dictaminar si la información que ha recibido el usuario del que se está estudiando la credibilidad presenta fuentes fiables y por tanto si la información es creíble.

El problema que tiene esta opción consiste en que las personas, por muy expertas que sean en un tema, no siempre dicen la verdad o no transmiten una información totalmente creíble. Por tanto no es posible establecer el grado de credibilidad de la información con estas tendencias.

8.1.3 Análisis de Metadatos

Debido a que los planteamientos anteriores no son efectivos para nuestra situación, se decide analizar diferentes formas de clasificación. Realizando la búsqueda de las tendencias actuales, se observa que los artículos [10] y [9] establecen que los estudios en torno a los tweets se deben de centrar en los metadatos que llevan asociados los mismos.

Por ello se decide realizar los estudios desarrollados a lo largo del capítulo 4. Al obtenerse unos resultados satisfactorios se decide realizar la implementación en el sistema. Con las diferentes pruebas realizadas en el

programa *Sniffer* (capítulo 7) se descubre que son satisfactorias en el primer nivel.

En el segundo nivel se plantea el problema de que no se consiguen las mismas prestaciones que las arrojadas por el estudio cuando se implementa únicamente el árbol de decisión. Por tanto, se decide plantear el segundo nivel de clasificación por medio de la misma estructura de clasificadores del primer nivel. Sin embargo a la hora de implementarlo se ha visto que se obtienen unas mejores prestaciones en el segundo nivel cuando no se utiliza un clasificador *MLP* por lo que se decide retirarlo y emplear los clasificadores basados en *KNN*, *árbol de decisión* y *Random Forest*.

8.2 Extracción y almacenamiento de tweets

En el momento de extraer información de Twitter se puede recurrir a diferentes librerías. Una de las librerías más utilizadas es *Tweepy* la cual aporta una interfaz sencilla en para la extracción de información. Sin embargo, a mi parecer esta librería se queda un poco corta y a la hora de solicitar una cantidad de datos grande lo estructura de tal forma que no es trivial la extracción de los contenidos. Como otra opción se presenta *TwitterApi*, la principal ventaja de esta API de Twitter es que presenta los datos en JSON e igual estructurados que como lo plantea Twitter por lo que es muy intuitiva a la hora de utilizarla y es por ello por lo que para la realización del programa se haya decantado por su utilización.

Como se ha tratado en otros capítulos (5.3, 6), una parte fundamental del programa *Sniffer* consiste en el almacenamiento de toda la información que se ha recabado. Inicialmente se planteó el almacenamiento de los resultados en archivos CSV. Esta idea se desechó muy pronto debido a que la gestión de los registros cuando el volumen de datos era muy grande ralentizaba mucho los procesos de consulta. De este modo se decidió recurrir al almacenamiento por medio de bases de datos. Así pues, se ha realizado la implementación de las bases de datos utilizando *SQLite3* pues es una herramienta que permite reducir la complejidad de las consultas y también el tiempo que se emplea, sea cual sea el tamaño de los datos.

Debido a que el programa tiene como base el almacenamiento de los datos, existe una tendencia a la formación de *"cuellos de botella"* en el acceso a los datos. Por este motivo, se comprobó cuál es el tipo de recursos

que más se utilizaba a la hora de analizar los tweets y se observó que eran los las consultas relacionadas con los usuarios. Con esto se decide cargar los datos de cada usuario que ha escrito un tweet en el contexto en la memoria del programa.

8.3 Planificación y presupuesto

8.3.1 Tareas realizadas

El modelo que se ha utilizado para la realización de este proyecto se basa en la arquitectura de cascada[1]. No obstante se ha realizado versiones semanales del sistema controladas en todo momento por la herramienta Git[2]. A su vez, semanalmente se ha realizado un informe reportando los avances del proyecto.

A lo largo de la ejecución del proyecto se pueden establecer diferentes tareas que se han cubierto. Estas tareas han sido las siguientes:

A Definición de objetivos del proyecto (15 Horas)

B Estudio del estado del arte (58 Horas)

C Planteamiento de la propuesta (73 Horas)

D Evaluación de la propuesta (28 Horas)

E Implementación de la extracción de tweets (35 Horas)

F Implementación del primer nivel de clasificación (49 Horas)

G Implementación del segundo nivel de clasificación (65 Horas)

H Implementación del algoritmo de obtención de altavoces (49 Horas)

I Ajuste de parámetros de los clasificadores (27 Horas)

J Desarrollo de pruebas del sistema (68 Horas)

K Evaluación del sistema (37 Horas)

L Implementación del programa en el servidor web (57 Horas)

[1] https://es.wikipedia.org/wiki/Desarrollo_en_cascada
[2] https://git-scm.com/

M Pruebas de funcionamiento del servidor (38 Horas)

N Evaluación de la implementación final (62 Horas)

O Documentación del Proyecto (160 Horas)

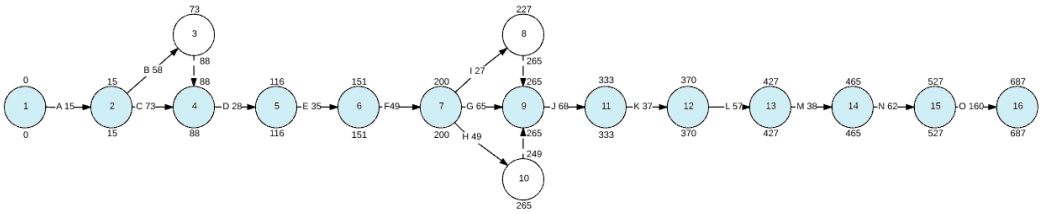

Figura 8.1: Diagrama PERT del proyecto

Como puede verse en el diagrama PERT (figura 8.1) la duración mínima del proyecto, basada en el criterio del camino crítico (marcado en azul), son 687 horas distribuidas a lo largo de 20 horas semanales, haciendo una duración total de aproximadamente 8 meses.

8.3.2 Estimación de costes

Se va a proceder a realizar una estimación del posible coste que tendría asociado la creación del sistema *Sniffer*.

Costes de personal

Para la realización de las tareas que se han mostrado en el apartado 8.3.1 se va a necesitar del personal que se dispone a lo largo de la tabla 8.1. El coste por horas de este personal es el que se dispone también en la tabla. El jefe de proyecto ha sido compuesto por el tutor del proyecto.

Cargo	Coste (por hora)
Jefe de Proyecto	45,00 €
Analista	30,00 €
Desarrollador	25,00 €
Gestor de Calidad	28,00 €

Tabla 8.1: Costes de Personal

Actividad	**Cargo**	**Núm. de horas**	**Coste (por hora)**
Definición de objetivos del proyecto	Jefe de Proyecto	15	45,00 €
Estudio del estado del arte	Analista	58	30,00 €
Planteamiento de la propuesta	Analista	73	30,00 €
Evaluación de la propuesta	Jefe de Proyecto	28	45,00 €
Extracción de tweets	Desarrollador	35	25,00 €
Primer nivel de clasificación	Desarrollador	49	25,00 €
Segundo nivel de clasificación	Desarrollador	65	25,00 €
Obtención de altavoces	Desarrollador	49	25,00 €
Ajuste de parámetros	Analista	27	30,00 €
Pruebas del sistema	Gestor de calidad	68	28,00 €
Evaluación del sistema	Jefe de Proyecto	37	45,00 €
Desarrollo del servidor web	Desarrollador	57	25,00 €
Pruebas del servidor	Gestor de calidad	38	28,00 €
Evaluación final	Jefe de Proyecto	62	45,00 €
Documentación	Desarrollador	160	25,00 €
Total		821	24.473,00 €

Tabla 8.2: Costes totales de personal

Aplicando estos costes a la distribución de horas que se ha realizado por parte de cada uno de los integrantes del proyecto, se obtiene el coste total asociado al personal del proyecto. Este coste es el que se representa en la tabla 8.2.

Por tanto se puede finalizar en que el coste total procedente del personal asciende a una cuantía de 24.473,00 € .

Costes materiales

Para el desarrollo del proyecto se han empleado diferentes materiales, estos materiales se pueden separar en equipos utilizados y licencias de programas utilizados.

Equipamiento

Para el correcto funcionamiento del programa *Sniffer* ha sido necesaria la utilización de un equipo que conste de los siguientes elementos:

- Procesador de 6 núcleos con una velocidad de trabajo de 3.4 GHz.
- Memoria RAM de 8 GB.

El equipo que se ha adquirido está valorado en 689 €

Licencias

Todos los programas que se han utilizado para el desarrollo tanto del estudio como del sistema *Sniffer* son de código libre. Por tanto, su coste total asciende a 0 €.

Costes Indirectos

En este apartado se van a introducir todos los costes indirectos que ha tenido asociados el desarrollo del proyecto. Estos costes están representados en la tabla 8.3. El total de la cuantía asciende a 2.861,00 €[3] [4].

Concepto	Coste
Alquiler del local	1.350,00 €
Luz y agua	154,00 €
Servicio de limpieza	160,00 €
Teléfono y acceso a Internet	137,00 €
Cobertura de bajas	840,00 €
Seguro a todo riesgo	220,00 €
Total	2.861,00 €

Tabla 8.3: Costes Indirectos

[3]El seguro a todo riesgo incluye cobertura sobre los bienes y sobre los trabajadores

[4]La cobertura por bajas se aplica en caso de tener necesidad de contratar personal para cubrir bajas por enfermedad o incapacidad.

Coste Total

A continuación se presenta el precio total del sistema. Hay que denotar que a este coste se le ha añadido el coste asociado a los impuestos (21 %) y el coste asociado al riesgo (18 %) y el margen de beneficios que se pretende generar (20 %). En total es lo que se representa en la tabla 8.4.

Concepto	**Cuantía**
Costes de personal	24.473,00 €
Costes de material	689,00 €
Costes indirectos	2.861,00 €
Riesgo (18 %)	5.044,14 €
Beneficios (20 %)	5.604,60 €
Total sin I.V.A	38.671,74 €
Total con I.V.A (21 %)	46.792,81 €

Tabla 8.4: Coste total del sistema

Por tanto el precio final del sistema *Sniffer* asciende a 46.792,81€ (**Cuarenta y seis mil setecientos noventa y dos euros con ochenta y un céntimos de euro**).

Capítulo 9

Conclusiones y trabajos futuros

En este capítulo se va a plantear una recapitulación de los principales conceptos de la aplicación, así como diferentes planteamientos que se podrían implementar de cara a futuras investigaciones o trabajos.

9.1 Principales conclusiones

Uno de los principales conceptos que debe ser entendido es el siguiente: La información que se transmite en los mensajes publicados en las redes de *microblogging* normalmente no está completa. Por tanto, es necesario utilizar contenidos adicionales (metadatos) que permitan conocer el contexto del contenido.

Una vez comprendido esto, se puede comenzar a analizar el contenido y plantear el ámbito de trabajo del sistema. Para este programa, el principal objetivo que se ha planteado ha sido la realización de un sistema que sea capaz de ayudar en la clasificación de tweets en base a su credibilidad en el contexto. Como parte fundamental de este proyecto se ha planteado que esta clasificación conste de dos pasos:

- Primer nivel de clasificación: Esta clasificación extrae si el contenido del tweet presenta información o no.
- Segundo nivel de clasificación: Esta clasificación extrae si la información que contiene el tweet es creible o no.

A la vista de las pruebas desarrolladas en el capítulo 7, el desarrollo de ambos procesos se pueden considerar como satisfactorio, teniendo en cuenta las limitaciones de tiempo y volumen de datos de entrenamiento. Esto hace

denotar que la utilización de los metadatos asociados a los tweets para clasificar el texto es una opción que siempre se debe de tener muy en cuenta.

Una de las grandes equivocaciones en los estudios que realizan un análisis en torno a los mensajes que se publican en las redes sociales, es que no se analiza a quien lo publica. Esta aproximación trata de aportar una gran parte del grado de credibilidad en función del usuario. Por tanto, es necesario analizar al autor del mensaje que se quiere analizar.

El programa *Sniffer* consta de un proceso de obtención de altavoces para cada contexto. Esta secuencia es la que presenta un mayor coste computacional debido a que se debe comparar cada texto con todos los demás. Por tanto, es la fase de la clasificación en la que se emplea la mayor parte del tiempo.

El tiempo que tarda en realizar esta tarea no es determinante, pues no es una parte clave de la clasificación. El objetivo principal de este proceso es solo simplificar la visualización de los resultados. Por esta razón es por la que no se ha tratado en exceso la opción de minimizar los tiempos de ejecución. Si se quisiese implementar la aplicación en un entorno de tiempo real, sí que se debería de prestar atención en el tiempo de ejecución de este proceso.

A simple vista, la elección de las variables relacionadas con el texto puede permitir que este sistema se pueda utilizar para diferentes idiomas. Los idiomas que este programa podría soportar deben tener los signos de cierre de interrogación y exclamación. Por tanto no se descarta que este programa tenga capacidad de operación en múltiples idiomas como Latín, Francés o Inglés.

Este proyecto únicamente pretende ayudar en la creación de una base sólida para futuros trabajos relacionados con la obtención de la credibilidad de fuentes sociales.

9.2 Trabajos futuros

Para finalizar, uno de los trabajos que se podrían realizar en un futuro con este programa sería el dotarle de una actividad en tiempo real. En este proyecto se ha planteado el estudio y la implementación de cómo se realizaría esta clasificación. Sin embargo; esta aplicación es muy útil si se está utilizando en tiempo real debido a que la gente no necesita saber si un tweet que ha sido publicado hace tres horas es creíble o no. La gente necesita conocer si se puede depositar la confianza en un tweet que se acaba de publicar. Por estas razones se considera que éste es uno de los trabajos más importantes que se puede realizar con el programa *Sniffer*.

Debido a que Twitter es una red global, interfieren diferentes idiomas. Por esta razón sería recomendable comprobar el funcionamiento del sistema planteado en diferentes idiomas, comenzando por inglés, y posteriormente con derivados del latín y anglosajones. Una vez realizada esta comprobación y detectando posibles fallos sería recomendable adaptar la obtención de las variables relacionadas con el texto a idiomas mayoritarios como chino o ruso.

Un trabajo que no se debe descartar nunca es ampliar el set de entrenamiento y con ello poder mejorar aún más las prestaciones del sistema. Esto se debe a que se ha trabajado con un set de entrenamiento muy limitado y por ello pueden existir limitaciones en la clasificación.

Glosario

AJAX Asynchronous JavaScript and XML, Javascript asíncrono y XML. 69, 127, 128

API (Application Programming Interface, Interfaz de programación de aplicaciones. 17, 18, 24, 51, 85

CSS Cascading Style Sheets, Hojas de estilo en cascada. 128

CSV Comma Separated Value, Valores separados por comas. 85, 138

DOM Document Object Model, Modelo de Objetos del Documento es un convenio para representar e interactuar con objetos en documentos HTML, XHTML y XML. 127, 128

HTML HyperText Markup Language. 127, 128

IoT Internet of Things, Internet de las cosas. 9

JS JavaScript. 127

JSON Notación de objetos de JavaScript. 51, 71, 85

KNN K-nearest neighbours, K-vecinos más cercanos. 37, 38, 41, 42, 44, 45, 62, 76, 77, 85, 112, 113

MLP MultiLayer Perceptron, Perceptrón multicapa. 37, 39, 41, 42, 45, 62, 76, 85, 112, 113, 133

PERT Project Evaluation and Review Techniques, Técnicas de Revisión y Evaluación de Proyectos. 87

RAM Random Access Memory, Memoria de acceso aleatorio. 89

SVM Support Vector Machines, Máquinas de soporte vectorial. 15, 125

URL Uniform Resource Locator, Localizador de recursos uniforme. 35, 43, 52, 56, 111–113

W3C World Wide Web Consortium. 128

Bibliografía

[1] (2016) Lista de redes sociales más importantes y más utilizadas. [Online]. Available: http://www.marketingandweb.es/marketing/lista-de-redes-sociales-mas-importantes/?utm_content=buffer9eed5&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer

[2] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges." *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[3] S. Ziegler. (2015) Iot and crowd sourcing. [Online]. Available: https://www.itu.int/net4/wsis/forum/2015/Uploads/S/227/Pres-SebastienZiegler-IoTLab-Mandat-International-25May2015.pdf

[4] P. Naur, *Concise survey of computer methods*. Petrocelli Books, 1974.

[5] D. Wang, L. Kaplan, T. Abdelzaher, and C. C. Aggarwal, "On credibility estimation tradeoffs in assured social sensing," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 6, pp. 1026–1037, 2013.

[6] C. C. Aggarwal and T. Abdelzaher, "Social sensing," in *Managing and mining sensor data*. Springer, 2013, pp. 237–297.

[7] S. A. Jaen, "Diseño e implementacion de un sistema para el analisis y categorizacion en twitter mediante tecnicas de clasificacion automatica de textos," 2012.

[8] B. Klein, X. Laiseca, D. Casado-Mansilla, D. López-de Ipiña, and A. P. Nespral, "Detection and extracting of emergency knowledge from twitter streams," in *Ubiquitous Computing and Ambient Intelligence*. Springer, 2012, pp. 462–469.

[9] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 675–684.

[10] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, "Tweetcred: Real-time credibility assessment of content on twitter," in *Social Informatics*. Springer, 2014, pp. 228–243.

[11] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. ACM, 2007, pp. 56–65.

[12] K. Lerman and R. Ghosh, "Information contagion: An empirical study of the spread of news on digg and twitter social networks." *ICWSM*, vol. 10, pp. 90–97, 2010.

[13] G. A. Betancourt, "Las máquinas de soporte vectorial (svms)," *Scientia et Technica*, vol. 1, no. 27, 2005.

[14] M. L. C. Martínez. (2016) Plataforma t-hoarder. [Online]. Available: http://t-hoarder.com/

[15] D. Hardt, "The oauth 2.0 authorization framework," 2012.

[16] J. D. ESTADO. (1999) Boe.es. [Online]. Available: http://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf

[17] A. B. M. Ruiz, "Convergencia y divergencia entre los tribunales del orden social y la agencia española de protección de datos en materia de control informático de la prestación de trabajo," 2012.

[18] (2016) Sitio web de twitter inc. [Online]. Available: https://twitter.com/privacy?lang=es

[19] T. I. Company. (2014) Política de servicio para desarrolladores. [Online]. Available: https://dev.twitter.com/es/overview/terms/policy

[20] (2015) Requisitos de visualización. [Online]. Available: https://dev.twitter.com/es/overview/terms/policy

[21] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[22] L. Yujian and L. Bo, "A normalized levenshtein distance metric," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1091–1095, 2007.

Appendix A

Introduction

This chapter provides an introduction to the context where this project is framed, describing the reasons which have motivated its realization and the objectives that are intended to reach. Later, there is presented a summary of the contents of each chapter and appendix that constitute this document.

A.1 Work motivation

Throughout history, the information has come through the called *traditional services*; these services have been television, radio and newspaper. However from the digital revolution, and more specifically with the emergence of smartphones, the society has begun to make use of the called social networks as their main source of information.

A social network consists in a structure composed by people, organizations and companies which are mutually connected by different type of relationship like friendship, kinship, economic or common interests. This term has been updated in the last few years pointing to a type of Internet site that favours the creation of virtual communities where it is possible to access services that allow users to create social groups according to their interests, sharing photos, videos and information in general. Some examples of different type of social networks are Facebook[1], Twitter[2] or Youtube[3].

Globally, there are plenty of social networks; however there are two that show a clear dominance above rest. Twitter and Facebook are leading the

[1]https://www.facebook.com
[2]https://twitter.com
[3]https://www.youtube.com

exponents of the shift toward new models of communication. Facebook can group more than 1.300 million of active users around the world [1]. However, this network is less attractive because typical information which is transmitted through this network has a personal nature. On the other side, Twitter is formed by more than 284 million active users and reaches a total of more than 600 million. Despite having a smaller number of users than Facebook, Twitter is interesting because it is the most important source of news because the most important informants at global level have a very active presence on this last social network. For that reasons it is possible to say that Twitter has become as one of the most important source of information on Internet.

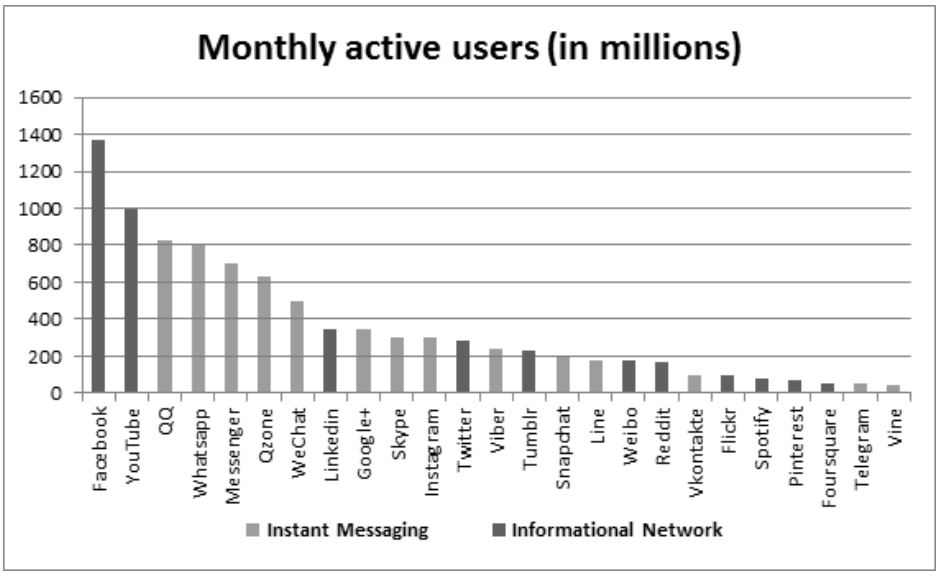

Figure A.1: Distribution of monthly active users along different social networks Source: *http://www.marketingandweb.es/*

As these types of platforms have obtained users, they have acquired a privileged position both in the creation and in the dissemination of news. On these platforms each user behaves as a source of news and the mesh that sets around him, whether friends (Facebook) or followers (Twitter); composes the elements of information distribution. This allows that information can be spread almost instantly along the network. This is the main reason why in periods of crisis, natural disasters and emergency situations; these channels are becoming crucial elements for the dissemination of useful information to affected users.

However, due to the enormous ease of dissemination of information on social networks appears a crucial problem. This problem occurs in the face of the enormous amount of information which is faced by the user. This information has been issued without any control of the users or administrators, unlike traditional media whose information is reviewed and contrasted with the purpose of transmit information as truthful as possible. Therefore, when user needs to obtain the information from this type of platforms, he needs that, previously; an information analysis was performed. This analysis allows users to associate a certain degree of credibility to every message. This analysis is helping users to discern above the veracity of the content, facilitating the detection of lies or false information which is propagated without any control along the network.

An example of what can be considered as credible information and less credible information are the screenshots from the social network Twitter which are represented in figure A.2. On these catches are shown two messages (Tweets). The first message can be considered as credible for different reasons: the user author is an official entity, he does not use foreign words, the text is written with all linguistic elements (tildes, commas) and also the post has a great impact on the social network: more than 7,700 rebroadcast (ReTweet) and more than 2,500 *Likes* (Favs). However, the second Tweet does not generate the same confidence than previous, either because we do not know anything about its author, the text has the presence of different foreign elements as the second label (hashtag) or its less relevance within the context, this tweet has not got any rebroadcast or any *Like*.

Figure A.2: Example of tweets

This task, which seems easy for people, is very complicated to perform with machines because the received information does not present any stable structure. Therefore this is what will be developed throughout this

document, a possible proposal for the automatic extraction of credibility of messages posted at Twitter. The main reason why it is wanted to be performed this task automatically, is that every day in Twitter more than 320 million of tweets of diverse nature are posted. Consequently, the user of this social network can not be checking constantly which tweets are truthful or credible, humanly this task is impossible. With the purpose of making the stay of the user in the social network as comfortable as possible, it is necessary to implement some tools which realize the above mentioned classification. With this tool, the user can directly check if the content of the message is credible or not credible within its context.

It should be noted this last which has been mentioned. It is not possible to determine if a content of a tweet is credible or not in isolation. If each Tweet of one context is considered individually, every relation between rest of tweets will be broken, so the information which contains these messages in the environment wheres they are posted will be lost. It is not the same that isolated messages are considered as credible when a user submits the information "There is an earthquake in Madrid" and when we consider it within its environment and the majority of users issue "There is not an earthquake in Madrid" and the context was started 3 days ago. Summarizing this, a message will not say the same thing when it is understood in isolation and when it is understood according to its context.

As it want to show along these lines, determine the credibility of the information which is being received, which is for what this work is intended, can be useful for:

- The capture or the modelling of the data of a draft analysis of the contents. It can be the first filter which allows to extract a set of optimum samples in the context where the study wants to work.
- Due to the receiver can be a governmental entity, it is possible that they are able to extract some really valuable information of social networks, for example, about certain incidents that may affect the life of a society to be shared with a certain degree of reliability.
- In situations of social emergency, to help the emergency services and affected in coordination tasks.
- To know the truest perception that different users of the social network have about the performances of a certain entity, people or enterprise.

A.2 Objectives

As explained previously, the main objective of this project will be the creation of a system which is able to help on the automatic classification of the credibility of different messages (tweets) posted by every kind of users at the social network Twitter. These messages will not be classified in isolation; the analysis will be performed by the context where they have been issued. This context will be referred throughout the project as *trend* (or *Hashtag* in the jargon of Twitter).

Throughout the following pages, there will be explained different stages that this project has. Each of these stages has its own goal, however; joining all of these partial objectives, the final objective of the project which has just been explained will be obtained. The partial objectives for the different stages are the following:

- To perform an analysis of each variables which can have a greater influence on the classification of the tweet. (Chapter 4)
- To perform the extraction and storage of tweets by the context where they have been posted. (Chapter 5)
- To perform the classification of the tweets in function of the variables above. (Chapter 5)
- To perform a system for the representation of the results in order to understand how it is the context where the tweet has been posted. (Chapter 6)

This system provides a complete operation, from messages collection (tweets) to the classification and representation of the results. Later, these objectives will be explained in more detail and its operation will be developed.

The question that this project helps to answer during the present document is: **How can credibility be obtained in Twitter?** As a solution this project proposes the system called *Sniffer*. This system is a contribution to the credibility problem that has been explained during this chapter.

A.3 Memory contents

This document constitutes the report of the project. It is structured in the following chapters:

- **Chapter 2**: This chapter will show different related works with this project. The situation of the different trends that exist will be explained and finally, a brief conclusion that the initial situation of the project will be drawn up.
- **Chapter 3**: This chapter will show usage limits for Twitter data. Those limits establish what we can do with these registers according to Spanish laws.
- **Chapter 4**: This chapter tries to explain a theoretical base between the relations of the variables from the metadata of the tweet for the later implementation in the *Sniffer* program.
- **Chapter 5**: This chapter tries to explain the structure and algorithms for classification part of *Sniffer* program.
- **Chapter 6**: This chapter tries to explain the structure and algorithms for visualization part of *Sniffer* program.
- **Chapter 7**: This chapter tries to explain the testing program for *Sniffer* system.
- **Chapter 8**: This chapter tries to explain different stages that this project has been passed.
- **Chapter 9**: This chapter will explain all conclusions for this project and it will propose future works.
- **Appendix B**: This chapter will explain a summary of the project and main milestones of it will be explained.
- **Appendix D**: This chapter will explain all technologies that *Sniffer* program uses.
- **Appendix E**: This chapter will show how to install *Sniffer* program.
- **Appendix F**:This chapter will show how to use *Sniffer* program.

Appendix B

Summary

This chapter will explain a summary of the project and main milestones of it will be explained.

B.1 Introduction

Throughout history, the information has come through the called *traditional services*, these services have been television, radio and newspaper. However from the digital revolution, and more specifically with the emergence of smartphones, the society has begun to make use of the called social networks as their main source of information.

In over the last few years, this term has been updated by pointing to a specific type of internet site which provides different services to create virtual communities according to the interests of the user. As these types of platforms have obtained users, they have acquired a privileged position both in creation and in the dissemination of news because each user behaves as a source of news and the mesh which is set around him, called friends (Facebook) or followers (Twitter); composes the elements of information distribution.

This allows that information can be spread almost instantly along the network. This is the main reason why in periods of crisis, natural disasters and emergency situations; these channels are becoming crucial elements for the dissemination of useful information to affected users. However, due to the enormous ease of dissemination of information on social networks appears a crucial problem. This problem occurs in the face of the enormous amount of information which is faced by the user. This information has been

issued without any control of the users or administrators, unlike traditional media whose information is reviewed and contrasted with the purpose of transmiting information as truthful as possible.

B.2 Main objectives

In order to provide a solution to the credibility problem, the main objective of this work has been the creation of a system which is able to help to the automatic classification of the credibility of different message (tweets) which has been posted by every kind of users according to their context at the social network Twitter. These messages, as has already been said; will not be classified in isolation; the analysis will be performed by the context where they have been issued. This context will be referred throughout the project as *trend* (or *Hashtag* in the jargon of Twitter).

Figure B.1: Classification sequence

Each stage of this project has its own goal, however; joining all of these partial objectives, the final objective of the project that has just been explained will be obtained. The partial objectives for the different stages are the following:

- To perform an analysis of the variables that have a greater influence on the classification of the tweet. (Chapter 4)
- To perform the extraction and storage of tweets by the context where they have been posted. (Chapter 5)
- To perform the classification of the tweets in function of the variables above. (Chapter 5)
- To perform a system for the representation of the results in order to understand how it is the context where the tweet has been posted. (Chapter 6)

The question that this project pretends to answer during the present document is: **How can credibility be obtained in Twitter?** As a solution this project proposes the system called *Sniffer*. This system tries to provide a helpful tool for the credibility problem at the social network Twitter.

B.3 Results

B.3.1 Metadata analysis

It is necessary to explain that each user at Twitter presents a different behaviour on its own communication policy. Therefore if all users are grouped on the same classification system, it is possible that classification performance can be deteriorated. This is the main reason why the set of users has been segmented in homogenous groups.

The option which has been proposed as a basis for this segmentation is the relationship between friends and followers. This relationship allows to perform the classification of users into 3 different groups:

- **Common users**: These users establish their communication policy on the interaction with the rest of the users and their information comes from different sources (friends). This group of users represents the biggest part of total users in Twitter (80%). Their ratio friends/followers is less than 2.

- **Corporative users**: These users establish their communication policy on the dissemination of information; however, they have a certain degree of interaction with users. This group of users has many companies, social media or celebrities and represents almost 11% of total users.

- **Organizative users**: These users only establish their communication policy on the dissemination of information. estos usuarios (ser consistente) This group of users contains different state entities like emergency services and represents almost 9% of total users in Twitter. These users have an enormous power of diffusion and influence.

As a result of this process of segmentation, the set of users is divided into three groups that are the ones that have been represented in the figure B.2. This classification is performed during the first phase (User Classification) represented on figure B.1.

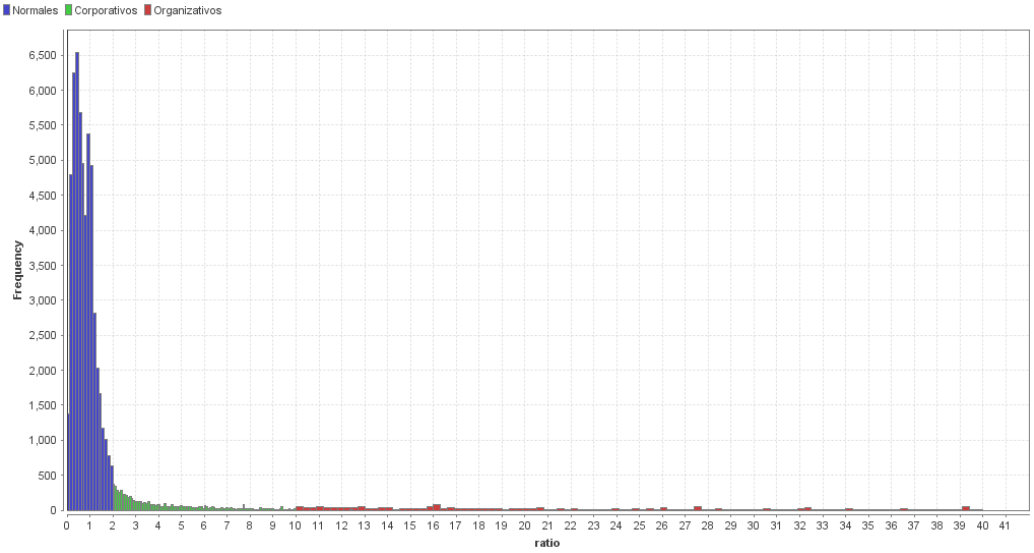

Figure B.2: User segmentation

In order to raise a possible solution for credibility problem it is neccesary to understand the environment which this project is situated. This is the reason why we analyze different parameters which have been used across different papers related with this problem ([9] and [10]). According to these two papers we are going to raise every parameter which will be analyzed in order to check the influence in the final decision of the program. These parameters are represented along table B.1.

According to these parameters, the classification structure which will be developed is the one that is represented in figure B.1. This sequence will be used during the study and development of *Sniffer* program.

After defining the classification sequence, it is necessary collect each tweet which will be used by the *Sniffer* program. These tweets will be used during training and validation phase for each classification level. For this classification task, it has been developed on the visualization module an option where user can add classified tweets to the training set of *Sniffer* system. The information which user can have during this classification process is very important because we are trying to simulate the same decision conditions of Twitter's application. In our case, the user will only have the following information:

- **Text**: This field contains the text of the tweet which will be analyzed.

- **Nickname**: This field contains nickname of tweet's author (@policia)
- **Class**: This field contains orientative information which proceeds from first classification level of the tweet.
- **Credible**: This field contains orientative information which proceeds from second classification level of the tweet.

Feature	**Description**
Length of tweet	Length of the text of the tweet, in characters
Number of words	Number of words of the text of the tweet
Number of monosyllable	Number of monosyllabic words of the text of the tweet
Number of *via*	Number of *via* contained on a tweet
Number of question mark	Number of question mark contained on a tweet
Number of exclamation mark	Number of exclamation mark contained on a tweet
Tweet's age	The relative age of the tweet since the beginning of context (in hours)
Number of URL's	Number of URL's contained on a tweet
Number of mentions	Number of mentions contained on a tweet
Number of hashtags	Number of Hashtags contained on a tweet
Location	Tweet contains location
Ratio tweets/followers	Relation betweetn number of statuses and followers
Ratio friends/followers	Relation between number of friends and followers
Verified user	Author has a "verified" account

Table B.1: Influential features on the classification of tweets

Once this task has been finished, the next step is the development of the first classification level. This level will use every parameter represented on table B.1 and it will classify each tweet according to following classes:

- **Related tweets with info** (R1): These are the tweets that we want to extract its credibility. They are tweets which are related and they transmit certain information.

- **Related tweets without info** (R2): These tweets may interest us, however they do not have any information about the context, despite being related to context.

- **Non-related tweets** (R3): As a result of that we are dealing with specific contexts, these tweets are not useful because they are not related to the topic that is being analyzed.

- **Skip tweets** (R4): Errors may arise during the process of capture of tweets. Therefore, this class will be used for errors and retweets.

The study of the influence of different variables in the final decision has been performed using *Watson Analytics*[1]. Watson shows which attributes have a greater influence on the classification. These attributes are *Number of URL's* and *Age of tweet*. This selection is valid if it is used as the classification algorithm a *decision tree* algorithm. Therefore, we are going to test the performance of this algorithm with a simulation of it.

The simulation results set us an average accuracy around 71,05%. However, we think that it is possible to improve that results. This is the reason why we simulate this classification level with *KNN*, *Random Forest* and *MLP* algorithms.

Each one of these algorithms works fine within a given user group, however when they work whith rest of groups they behave worse. This is why it has been decided that all algorithms will work together. With this system we get a better performance than using only the decision tree, placing us on a 77, 38%. That is why it is decided to use the classifier based on multiple algorithms. The classification results of this level are represented on table B.2.

As we can see on table B.2, *Random Forest* algorithm has a better average accuracy than the classification system that we propose. However, *Random Forest* has a greater variation between each class of users. That is the reason why we choose the classification system based on four algorithms.

All posts that have been classified as R1 (Related and contains information) have access to the second classification level of the system. The reason for selecting these messages classified as R1, is that it is unnecessary to examine the credibility of the messages which contain no information. This second classification level pretends to analyse the credibility of each Tweet

[1]http://www.ibm.com/analytics/watson-analytics/us-en/

Algorithm	Common Users	Corporative Users	Organizative Users	Average Accuracy
Decision tree	72,16%	69,57%	71,42%	71,05%
KNN	62,89%	79,15%	89,66%	77,23%
Random Forest	69,07%	78,26%	89,66%	78,99%
MLP	64,95%	73,91%	93,10%	77,32%
All	70,95%	82,61%	78,56%	77,38%

Table B.2: First classification precission performances

which contains information. Each tweet will be classified according to these classes:

- **Credible** (C1): These tweets are those which have the greatest degree of credibility. They present a greater priority on this classification, because they are the most important messages in the context.
- **Less credible** (C2): These tweets have lower credibility degree than tweets classified as C1. The information that these messages are transmitting has to be interpreted carefully by the user because there is no guarantee of its total credibility.
- **Non-credible** (C3): These tweets can be considered as they have not got any credible basis within the context. That is the main reason why they should be discarded on every data collection.

The realization of this classification process has followed the same methodology than the one which has been carried out at the first classification level. Initially, there has been performed an analysis of the most influential variables in the decision with Watson Analytics. This analysis shows that, implementing a *decision tree* algorithm, the most influential variables on this clasification level are *Number of URL's* , *Ratio Tweets/followers* and *Location*.

According to the results obtained by Watson, it was decided to simulate the decision tree to obtain its performance. The obtained results are represented along table B.3. Due to the fact that the other level has worked well with the usage of the set of classifiers, it was decided to perform an analysis of same algorithms than first classification level. However, we discovered that no algorithm is able to improve the performance of the decision tree, so it was decided to implement only the tree (however, this

implementation in Python will change).

Class	Common Users	Corporative Users	Organizative Users
Credible	100% (3)	66,67% (6)	86,67% (15)
Less Credible	85,11% (47)	81,82% (11)	66,67% (9)
Non-Credible	100% (5)	0% (0)	0% (0)
Resolution	87,27%	76,47%	79,17%

Table B.3: Results of accuracy (number of elements) from the second classification level (Decision tree)

B.3.2 *Sniffer* program

Once completing the previous study we are able to understand the actual situation of the credibility problem. As a possible solution for this problem, it has been proposed the creation of the program called *"Sniffer"*. This system pretends to provide a complete functionality in the process of the analysis of tweets, from the extraction of tweets to their subsequent classification. The sniffer program presents a structure such as the one explained in the figure B.3.

As we can see on this figure, databases are one of the most important parts of the program. The reason of this importance is that we must to minimize the number of queries to Twitter. The main reason of it is that Twitter imposes a temporarily limitation in the number of queries for each application. The main database is *database.db*, this database stores four different tables:

- **Tweets**: This table stores all necessary content associated with each Tweet.
- **Train_Tweets**: This table stores all contents classified of each Tweet for training.
- **Users**: This table stores the data associated with each owner of each Tweet.
- **Credentials**: This table stores every credential necessary to establish communication with Twitter.

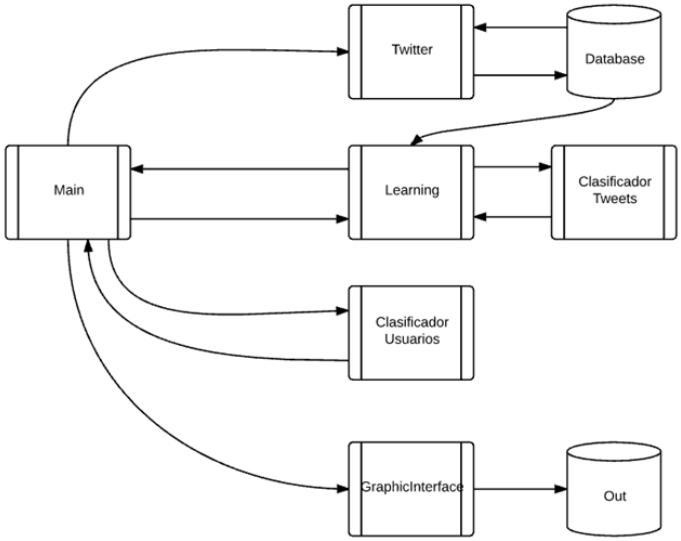

Figure B.3: Structure of *Sniffer* program

This second classification level is proposed as a result of metadata study presents a structure based only on a *Decision tree* classifier. However, during test phase, this structure presents poor precision performances for each class. This is the main reason why it is decided to use a structure based on *Decision tree*, *KNN* and *Random Forest* classifiers to improve this performances. These classifiers will act as first classification level structure.

Due to the fact that the program is working at both levels with multiple classifiers, it is necessary to select a library which is able to provide each required classifier. *Sklearn* [21] allows to obtain the probability of each decision of each classifier. In the decision between all classifiers is going to use *Sklearn* and, additionally; there is going to make a positive reinforcement to each classifier. In other words, it is going to weigh the likelihood that approximates each classifier with the success rate obtained during the training process. This process is the one that is represented in the figure B.4.

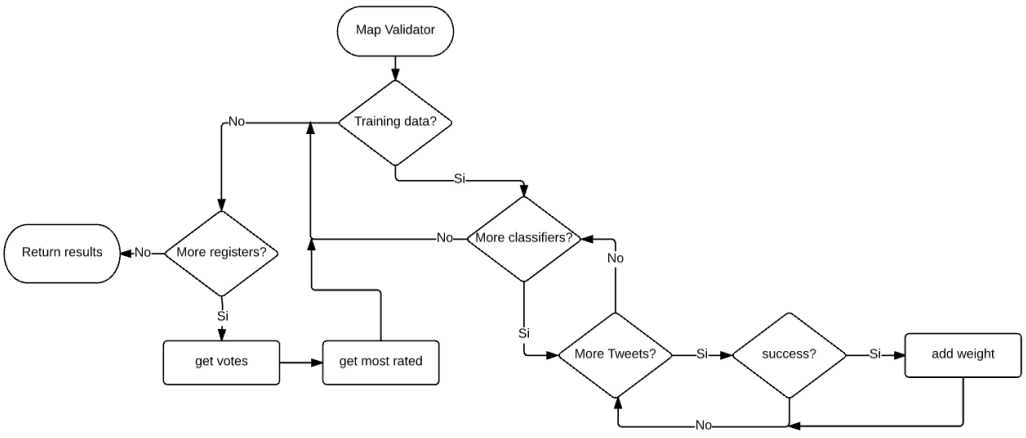

Figure B.4: Decision sequence

B.4 Results visualization

The representation of the results of the program is carried out on a web server, specifically on *apache-tomcat-8.0.28*. For each hashtag or context which has been analyzed, the *Sniffer* program creates three different views:

- **Pie chart**: This view represents the distribution of each class which exists in the classification.
- **Evolution chart**: This view represents the temporal evolution of the Tweets posted grouped by their class.
- **Tables**: This view contains all the results that have been used for the analysis and the creation of the other two views.

In order to simplify the visualization of *Sniffer* results, it has been developed a process which obtains the relevance of each tweet within its context. This procedure allows the user to get which tweets have a greater impact within the context. This procedure is called *"Obtaining Speakers"* and it has not got any effect on the classification proccess. This procedure can determine which tweets have a greater impact within the context and therefore allows users to access quickly to them.

Some used data can be considered as personal data, the reason is that it is possible to relate the content which is wanted to display with a physical person. These data in particular are the user identifier and the identifier of the tweet, because they are unique data on Twitter.

For this reason, it must be applied to these data an unidirectional transformation. Hash functions are used to perform this transformation. However, from the resulting value of this transformation is possible to obtain the original value, so previously it must be applied a *XOR* function between represented data and a secure generated random number.

B.5 Conclusions

One of the main concepts which must to be understood is the following: The information that is transmitted on messages posted on *microblogging* networks are usually not complete. Therefore, it is necessary to use additional contents (metadata) that enables us to know the context of the content.

As it has been seen throughout the project, the element which has got a crucial importance has been the context where the content of the tweet has been issued. Another element which must be understood is that in matters of credibility, truthfulness or falsehood, always is necesary to contrast the information which is being analyzed. That is the reason why the *Sniffer* program stores each tweet by its context, also called *hashtags*.

As a differentiating element from rest systems which have been raised, the Sniffer program employs user related parameters as an additional element on content classification. This procedure is allowing constrain the behavior of each user on each tweet, and improve the performance of the classification of content.

Appendix C

Conclusions and future works

This chapter will explain all conclusions for this project and it will propose future works.

C.1 Main conclusions

One of the main concepts which must to be understood is the following: The information that is transmitted on messages posted on *microblogging* networks are usually not complete. Therefore, it is necessary to use additional contents (metadata) that enables us to know the context of the content.

As soon as this concept is understood, we are able to begin to analyse the content and consider the scope of work of the system. For this program, the main objective that is proposed is the creation of a system which is able to help on the classification of tweets by their credibility according to the context that they have been issued. The most important process on this program is the classification sequence; this sequence is proposed on two steps:

- First classification level: This classification extracts if a tweet contains information or not.
- Second classification level: This classification extracts if the information which is contained on a tweet is credible or not.

According to the results from tests made during chapter 7, the development of both processes can be considered satisfactory according to limitations of time and volume of dataset that this project has. These results denote that the usage of the metadata associated with the Tweet is

one of the best options when we want to classify short length texts.

One of the biggest mistakes that exists in studies which carried out an analysis around messages which are posted on the social network, is that usually there is not any analysis about who publishes it. This approach aims to provide a significant part of credibility degree of the tweet in function of user analytics. So it is necessary to analyse the author of the message which wants to be analysed.

The *Sniffer* program has a process which obtains speakers for each context. This sequence represents the main part of computational cost of the program because it must to compare every text with rest. So this phase will spend most of time.

The time that is spent by the program to perform this task is not decisive because it is not part of the classification process. The objective of this sequence is only for visualization. For this reason it is not optimized the time that is wasted on this task. However, if future works want to upgrade this application to a real-time environment, they should improve the execution time of this task.

At first sight, the choice of the variables related with the text allows this system to be used for different languages. The languages which this program can analyse must have interrogation and exclamation signs of closure. Therefore it can not be ruled out that this program is able to operate over multiple languages like Latin, French or English.

This project has been developed to help in the creation of a solid basis for future works related to the obtaining of the credibility on social-sensing environments.

C.2 Future works

Finally, one of the related works that could be carried out in future with this program could be to provide a real-time activity for the *Sniffer* system. Throughout this project it has been explained the study and a possible implementation for the classification model which has been proposed. However, this application improves its usage if users use it in real time, because people do not need to know if a tweet which has been posted 3 hours ago is credible or non-credible. They need to know if a tweet that is posted now is credible or not. For this reason it is possible to believe that this work is one of the most important options that are possible to do with *Sniffer* program.

Due to Twitter is a global social network there are converging different languages. For this reason, it is recommended to check the operation of *Sniffer* program with different languages at the same time like English, French or Italian. As soon as these checks are realized and different faults are solved, it should be advisable to adapt text related parameters to non-Latin or non-Anglo-Saxon languages like Russian or Chinese.

A work that we must never discard is to improve the training set. With this improvement is possible to increase even more the performances of the system. This work is proposed because the training set which is used is very limited and is possible that do not appear different problems with this set.

Apéndice D

Tecnologías Utilizadas

En este capítulo se va a tratar de explicar cada una de las tecnologías y herramientas que se han utilizado en el desarrollo del proyecto "Sniffer".

D.1 Watson Analytics

Watson Analytics es una plataforma creada por IBM en su afán por liderar el rumbo de la investigación en el campo de la inteligencia artificial. Esta plataforma se utiliza en este programa con la finalidad de analizar las relaciones que existen entre los distintos parámetros y las diferentes clases finales del proceso de clasificación.

Figura D.1: Logo de Watson Analytics

D.2 Rapidminer

RapidMiner es un programa informático para el análisis y minería de datos. Permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico. Se usa en

investigación, educación, capacitación, creación rápida de prototipos y en aplicaciones empresariales. RapidMiner proporciona más de 500 operadores orientados al análisis de datos, incluyendo los necesarios para realizar operaciones de entrada y salida, preprocesamiento de datos y visualización. En este proyecto esta plataforma se ha utilizado para la realización de las simulaciones de los distintos sistemas de clasificación.

Figura D.2: Logo de RapidMiner

D.3 Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, y es multiplataforma. Este lenguaje de programación es muy utilizado para el prototipado rápido. La versión que se ha utilizado corresponde a la 2.7, este lenguaje es en el que se ha implementado el programa *Sniffer* debido a que permite una programación sencilla y es utilizado por una gran parte de la comunidad para la creación de plataformas de análisis de datos.

Figura D.3: Logo de Python

D.4 Scikit-learn

Scikit-learn es un software libre que permite la creación de sistemas de aprendizaje máquina para el lenguaje de programación Python. Sus capacidades abarcan algoritmos de clasificación, regresión y agrupamiento. Esta librería está principalmente escrita en el lenguaje Python, sin embargo algunos algoritmos están escritos en Cython como son los algoritmos correspondientes a las máquinas de soporte vectorial (SVM). Esta librería es la que se va a utilizar para la implementación de los diferentes clasificadores que componen el sistema de clasificación del programa *Sniffer*.

Figura D.4: Logo de ScikitLearn

D.5 Twitter Api

Esta librería es un software libre desarrollado por Mike Verdone. Twitter Api permite poder establecer comunicaciones con los servicios de Twitter para extraer la información necesaria. Esta librería en el momento en el que se ha desarrollado el proyecto se encuentra en su versión 1.17.1

D.6 Levenshtein

Esta librería es un software libre el cual permite poder calcular rápidamente la citada distancia de *Levenshtein* para hallar la similitud entre los diferentes contenidos de los tweets de un contexto.

D.7 SQLite3

SQLite es un sistema de gestión de bases de datos relacional cuyo motor, a diferencia de los sistema de gestión de bases de datos cliente-servidor, no es

un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

En su versión 3 (SQLite3), SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB.

Figura D.5: Logo de SQLite3

D.8 Apache

Apache Tomcat comúnmente conocido como Apache, es el servidor web HTTP empleado para servir la visualización del sistema *Sniffer* por medio de una aplicación web. Se trata de uno de los proyectos de software libre más veteranos que hospeda la Apache Software Foundation y destaca por su robustez, estabilidad y grandes posibilidades de configuración que unido a su licencia abierta y apta para entornos comerciales y de producción, hacen que sea el servidor web empleado por millones de sitios en todo el mundo, existiendo versiones para múltiples sistemas operativos y arquitecturas.

Apache Tomcat funciona como un contenedor de *Servlets*. Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Figura D.6: Logo de Apache Tomcat

D.9 JavaScript

JavaScript (también llamado JS) es un lenguaje de programación interpretado, derivado del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM). Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Figura D.7: Logo de JavaScript

JavaScript permite la introducción de diferentes librerías que facilitan la programación al usuario. Las librerías que se han utilizado para la realización del sistema de visualización del programa *Sniffer* son las que se exponen a continuación.

D.9.1 JQuery

jQuery es una librería de JavaScript que permite interactuar con los documentos HTML, manipular el árbol DOM, gestionar eventos, animar elementos y aprovechar todas las posibilidades que ofrece AJAX de una manera rápida y sencilla, siendo compatible con todos los navegadores y soportando los últimos estándares web aprobados por el W3C (*World Wide Web Consortium*). Al igual que otras librerías, jQuery ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Figura D.8: Logo de jQuery

D.9.2 Bootstrap

Bootstrap[1] es un framework de código para el desarrollo web en HTML5 creado por Twitter y mantenido por la comunidad, aunque su desarrollo principal sigue estando a cargo de miembros de esta red social.

Consiste en un conjunto de recursos web entre los que se incluyen CSS con estilos predefinidos, imágenes y librerías JavaScript que pueden ser personalizados y modificados según las necesidades de cada usuario, permitiendo desarrollar interfaces web de forma rápida y sencilla.

En nuestro proyecto se ha empleado Bootstrap en su versión 3.3.6 para construir la interfaz web a través de la cual los usuarios visualizan los resultados de los experimentos.

[1] http://twitter.github.com/bootstrap/

D.9.3 Morris.js

Morris.js[2] constituye una librería, la cual se sustenta sobre JQuery, y que permite al desarrollador la implementación de diferentes tipos de gráficos de una manera sencilla, rápida y versátil.

En el proyecto *Sniffer* se ha utilizado esta herramienta en su versión 0.5.1 para construir los diferentes gráficos que se han mencionado a lo largo del apartado destinado a la visualización de los resultados del sistema (Capítulo 6).

[2]http://morrisjs.github.io/morris.js/

Apéndice E

Manual de instalación

A lo largo de este capítulo se describen los pasos necesarios para la realización del despliegue del sistema Sniffer en una máquina, instalando y configurando el software necesario para el correcto funcionamiento.

E.1 Dependencias

Como se ha podido ver a lo largo del apéndice D el sistema requiere de una serie de tecnologías específicas para su funcionamiento, sin embargo estas tecnologías requieren de software específico para funcionar. Las tecnologías específicas que requiere el sistema *Sniffer* para funcionar son las siguientes:

- Scikit-Learn
- TwitterApi
- Levenshtein

E.2 Instalación

Existen multitud de formas de instalar estas todos estos productos, debido a que al tratarse de software libre y gratuito, habitualmente se encuentran en los repositorios del sistema operativo en el caso de distribuciones GNU/Linux, aunque también se encuentran en repositorios de Python enlazados a través de las herramientas *pip* o *easy_install*.

La configuración que se expone a continuación está realizada para una máquina que está ejecutando Ubuntu Linux 14.04 LTS o superior. Para nues-

tra instalación, se ha alternado entre la herramienta *pip* y el gestor de paquetes de la distribución.

E.2.1 PIP

pip Es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python incluidos en un directorio central. Esta herramienta constituye una alternativa a *easy_install* y presenta la posibilidad de incorporar nuevos repositorios a los existentes por defecto.

La instalación de esta herramienta se lleva a cabo mediante el gestor de paquetes de Ubuntu:

```
/$ sudo apt-get install python-pip
```

Figura E.1: Instalación de *pip*

E.2.2 Scikit-Learn

Esta librería desarrollada para Python permite al desarrollador la posibilidad de implementar clasificadores de aprendizaje automático de forma sencilla y con un amplio margen de maniobra, pudiéndose configurar multitud de parámetros de cada clasificador.

Para la instalación de esta librería es necesario disponer de otras dos librerías. Estas dos librerías son *Numpy* y *SciPy*. La instalación de estas dos librerías secundarias se ha realizado mediante la herramienta *pip* por medio de los siguientes comandos:

```
/$ sudo pip install numpy
/$ sudo pip install scipy
```

Figura E.2: Instalación de *numpy* y *scipy* (Opción 1)

Otra opción es utilizar el gestor de paquetes de Ubuntu:

```
/$ sudo apt-get install python-numpy
/$ sudo apt-get install python-scipy
```

Figura E.3: Instalación de *numpy* y *scipy* (Opción 2)

Para la instalación de la librería *Scikit-Learn* se está haciendo uso de la versión de desarrollador debido a que es necesaria la implementación de los clasificadores MLP. Actualmente esta versión de desarrollo es la 0.18, para su instalación es necesario seguir el siguiente comando:

```
/$ sudo pip install git+git://github.com/scikit-learn/
scikit-learn.git
```

Figura E.4: Instalación de *scikit-learn* (Opción desarrollo)

No obstante, si ya estuviese disponible la versión 0.18 de la librería de forma estable, la instalación se realizaría por el siguiente modo:

```
/$ sudo pip install scikit-learn
```

Figura E.5: Instalación de *scikit-learn* (Opción estable)

E.2.3 TwitterApi

Esta librería va a ser la que va a permitir realizar la comunicación con los servicios de Twitter. Para su instalación se va a hacer uso de la herramienta *pip* ejecutándose el siguiente comando:

La versión con la que se ha desarrollado el sistema *Sniffer* ha sido la identificada por 1.17

```
/$ sudo pip install Twitter
```

Figura E.6: Instalación de *TwitterApi*

E.2.4 Levenshtein

Esta librería va a permitir calcular la ratio asociada a la distancia de *Levenshtein* de una forma sencilla. Para su instalación se ha recurrido a la herramienta *pip* ejecutándose el siguiente comando:

```
/$ sudo pip install python-levenshtein
```

Figura E.7: Instalación de *Levenshtein*

E.3 Despliegue

Una vez todas estas librerías están instaladas, se relaiza la descompresión del archivo ZIP que contiene el servidor apache y el programa *Sniffer* en su interior.

Para facilitar la instalación de todas las librerías al usuario, se dispone un instalador (*make*) que se encarga de instalar todas y cada una de las librerías necesarias para el funcionamiento y al finalizar esta instalación descomprime toda la estructura propia del programa *Sniffer*. Este instalador permite cuatro acciones:

- **Instalar**: Permite instalar todas las librerías necesarias para el sistema *Sniffer* de forma automática.
- **Descomprimir**: Permite descomprimir todo el sistema de archivos propio del sistema *Sniffer*.
- **Comprimir**: Permite comprimir y eliminar toda la estructura de directorios que posee el sistema *Sniffer*.
- **Borrar**: Permite eliminar la estructura de directorios que aloja el sistema *Sniffer*.

```
/$ sudo make install
/$ sudo make unzip
/$ sudo make zip
/$ sudo make clean
```

Figura E.8: Despliegue automático (Opciones)

Para conocer el manejo del sistema se debe recurrir al Manual de Usuario (Apéndice F) donde se expone el funcionamiento tanto del sistema *Sniffer* como de su sistema de visualización.

Apéndice F

Manual de usuario

A lo largo de este capítulo se va a tratar de exponer como manejar el programa "Sniffer"por parte del usuario.

F.1 Sistema *Sniffer*

Inicialmente se pueden dividir los modos de funcionamiento del programa *Sniffer* en dos:

- **Modo *Normal***: Este modo es el normal de funcionamiento, está destinado a ser útil para todos los usuarios y no representa información más allá de la que es necesaria para comprobar el estado del programa.
- **Modo *Prueba***: Este modo está destinado a ser utilizado por los desarrolladores. Este modo actúa únicamente sobre la opción de estudio que más adelante se expone. La peculiaridad que añade este modo es que permite al usuario comprobar cuáles han sido los resultados del entrenamiento de los diferentes clasificadores.

El funcionamiento normal del programa no necesita de ningún parámetro para su ejecución, sin embargo para activar el *modo Prueba* es necesario aportar el argumento *"-p"* a la hora de ejecutar el programa.

Al ejecutar el programa *Sniffer* al usuario se le ofrece un menú con las diferentes funcionalidades que posee el programa. Estas funcionalidades son las siguientes:

- **Recogida**: Este modo de operación se encarga de realizar la recolección de los diferentes tweets. Se consulta a Twitter cuáles son los Hashtags disponibles, en nuestro caso en España; y se ofrece al usuario la opción

de seleccionar uno de esos contextos o aportar él mismo el que desee. Una vez establecido el Hashtag se empieza a recolectar los tweets.

- **Estudio**: Este modo de operación se encarga de realizar el experimento sobre cada uno de los contextos de los que se han recolectado tweets.
- **Exportación**: Este modo de operación permite al usuario la posibilidad de exportar los conjuntos de entrenamiento de cada nivel de clasificación y los conjuntos que se han utilizado para el experimento en tres documentos CSV.
- **Eliminar Hashtag**: Este modo de operación permite al usuario eliminar contextos del experimento y por tanto de la base de datos.
- **Actualizar base de datos**: Este método permite al usuario actualizar las localizaciones de los usuarios que, en el momento del registro de su tweet asociado; no se disponía de la localización.

El menú que se ofrece al usuario para distribuir la actividad del programa es el que se representa en la figura F.1. Como se puede ver se dan las diferentes opciones de funcionamiento. La importación de las librerías necesarias para el funcionamiento de cada uno de los diferentes modos de operación se realiza una vez se selecciona la opción.

```
alberto@alberto-VirtualBox:~/       /apache-tomcat-8.0.28/webapps/sniffer/Sniffer$ python Sniffer.py
Bienvenido a Sniffer
Selecciona una de las operaciones
Estudio => 1
Recogida => 2
Exportar => 3
Eliminar Hashtag => 4
Actualizar BD => 5

Introduzca el numero de accion (0 para salir):
```

Figura F.1: Captura del Menú del programa Sniffer

A continuación se van a exponer de forma detallada cada uno de los modos de operación que tiene el programa.

F.1.1 Recogida

Una parte muy importante dentro del sistema es la recolección de los tweets o mensajes. Para facilitar este proceso se decide de incluir una opción dentro del sistema. Esta opción ofrece al usuario la posibilidad de seleccionar uno de los Hashtag que se encuentran disponibles para España o bien introducir el que él mismo desee.

Un ejemplo de ejecución de esta secuencia está representado en la figura F.2.

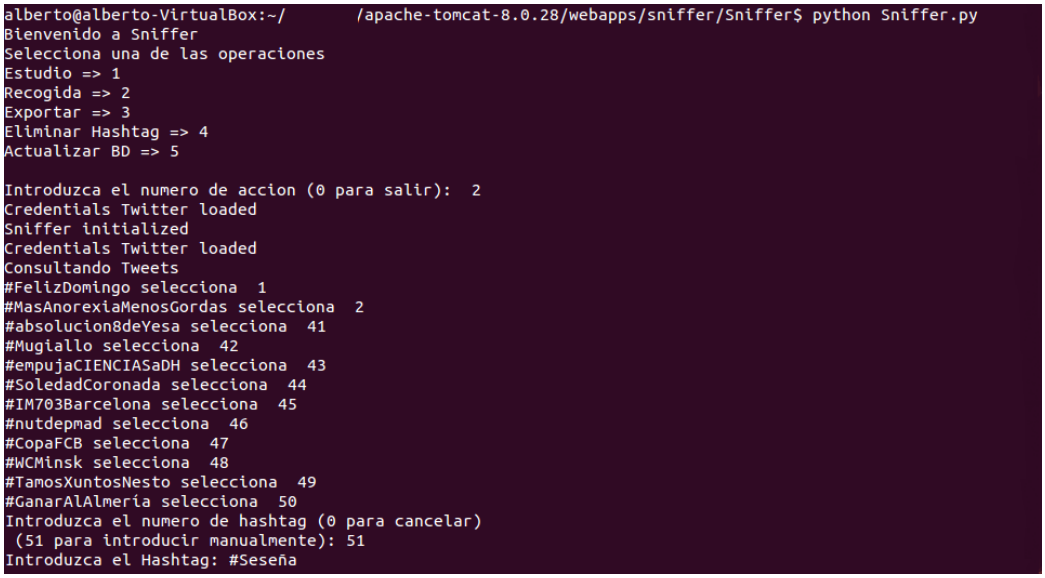

Figura F.2: Captura de la Recogida de tweets

Una vez se selecciona el Hashtag, se comienza a recolectar los tweets hasta que el usuario decide parar presionando el comando CTRL+C donde se vuelve al menú de Hashtags para volver a seleccionar o volver al menú principal de la aplicación.

F.1.2 Estudio

Este modo de operación es el principal de la aplicación. Se encarga de realizar la clasificación de la credibilidad de los tweets conforme se ha expuesto a lo largo de los capítulos 4 y 5. Como se ha comentado este modo de operación presenta dos tipos de funcionamiento, el modo *normal* y el modo *prueba*.

Al ejecutar el modo de prueba, lo que se consigue es obtener la estimación sobre la probabilidad de error que presentan cada uno de los clasificadores. Esta probabilidad se obtiene a partir del conjunto de entrenamiento de los diferentes sistemas de clasificación para los dos niveles de clasificación.

Un ejemplo de la ejecución de esta funcionalidad es el que se representa en la figura F.3 donde se ven los dos modos de ejecución (*normal* y *prueba*).

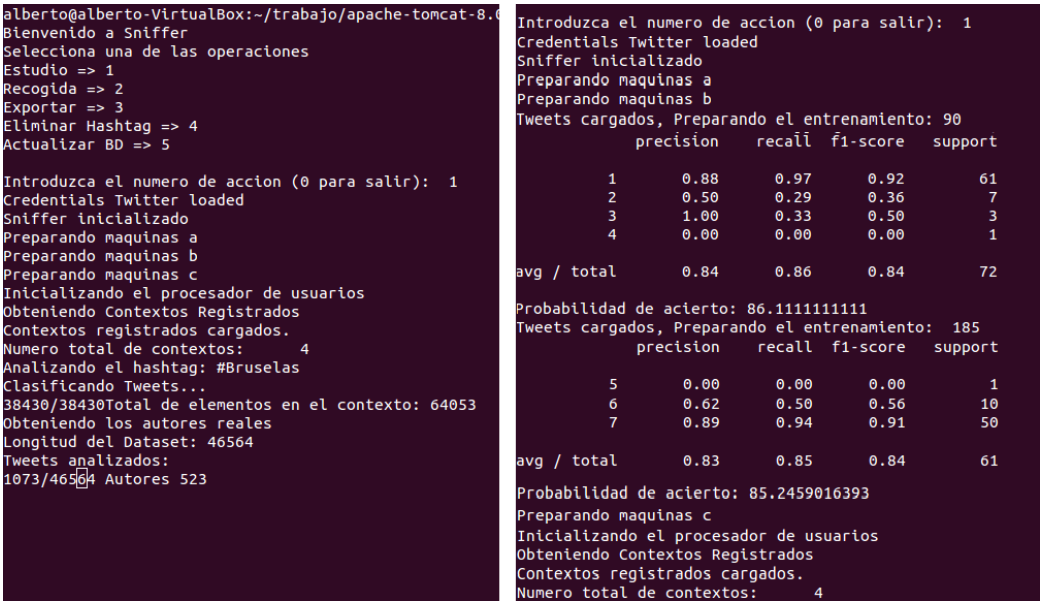

Figura F.3: Captura del modo Estudio (Izquierda modo normal, derecha modo prueba)

F.1.3 Exportación

Este modo de operación permite al usuario extraer los tweets que se utilizan para el entrenamiento de los clasificadores y lo que se emplean en el experimento. Esta acción genera tres ficheros en los cuales se almacenan los tweets de entrenamiento del primer nivel de clasificación, segundo nivel de clasificación y experimento respectivamente.

La finalidad de este sistema es poder proporcionar al usuario una mayor facilidad para poder realizar el estudio de los contenidos que se están utilizando con herramientas externas como se ha realizado en el capítulo 4.

F.1.4 Borrar Hashtag

Este modo de operación permite al usuario seleccionar un contexto, de entre todos los contextos registrados; para que sea eliminado. Para ello se mostrará un menú con los contextos que se tienen registrados para el experimento y posteriormente se deberá seleccionar el que se desee eliminar. Este proceso es el que se representa a lo largo de la figura F.4.

Con esta acción lo que se realiza es la eliminación de la base de datos de

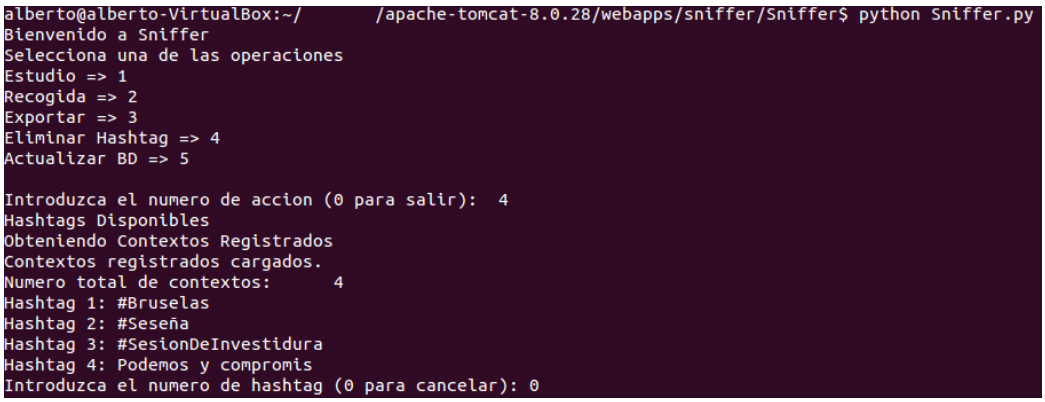

Figura F.4: Captura del modo Eliminar

todos los tweets relacionados con el contexto que se selecciona.

F.1.5 Actualizar base de datos

Al ejecutar este modo de operación lo que se está consiguiendo es que los usuarios cuyas localizaciones no están disponibles una vez se han recogido los tweets, se extraigan y actualicen los registros correspondientes. Esta pérdida de la localización puede ser debida a que se han capturado mal los datos o a que no estaba disponible en el momento de su recolección.

F.2 Visualización de Resultados

Para la ejecución del servidor *Apache* que contiene el sistema de visualización del programa *Sniffer* es necesario ejecutar el archivo *startup.sh* situado en la carpeta *bin* de la raíz de directorios del sistema.

Dentro de la visualización de los resultados del programa se presentan una serie de vistas, las cuales son las siguientes:

- **Inicio**: Es la vista principal del programa, se encarga de distribuir hacia el resto de las vistas que posee el sistema. Esta vista está representada en la figura F.5.
- **Requisitos**: Esta vista contiene todos los requisitos de librerías que contiene el sistema *Sniffer* y que son necesarios para su funcionamiento.
- **Feedback**: Esta vista contiene un pequeño script que permite al usuario aportar tweets al conjunto de entrenamiento del sistema *Sniffer*.

- **Contexto**: Esta vista pretende mostrar un pequeño resumen de lo que va a poder observar el usuario en el resto de las vistas asociadas a cada contexto.

Figura F.5: Vista Principal

A cada una de estas vistas se accede por medio del menú que figura en la barra de navegación de la vista. En este menú se establece una lista con todos los contextos que se han analizado en la ejecución.

A continuación se expone un ejemplo de cada una de las diferentes vistas asociadas a cada contexto y las vistas de requisitos y *Feedback* del sistema de visualización.

F.2.1 Contexto

Dentro de la visualización de los resultados de cada contexto, se presentan una serie de vistas. Estas vistas son las siguientes:

- **Distribución**: Esta vista muestra el conjunto de diagramas circulares mencionados anteriormente. En esta vista se representan exactamente cinco diagramas circulares los cuales están asociados a *"primer nivel de clasificación"*, *"altavoces altos"* , *"altavoces medios"* , *"altavoces bajos"* y *"no altavoces"*.

- **Evolución**: Esta vista muestra el conjunto de diagramas temporales mencionados anteriormente. En esta vista se representan exactamente

cinco diagramas temporales los cuales están asociados a *"primer nivel de clasificación"*, *"altavoces altos"* , *"altavoces medios"* , *"altavoces bajos"* y *"no altavoces"*.

- **Datos**: Esta vista muestra los resultados que ha arrojado del sistema con cada uno de los tweets del contexto. En esta vista se representan exactamente cinco tablas de datos las cuales están asociadas a *"primer nivel de clasificación"*, *"altavoces altos"* , *"altavoces medios"* , *"altavoces bajos"* y *"no altavoces"*.

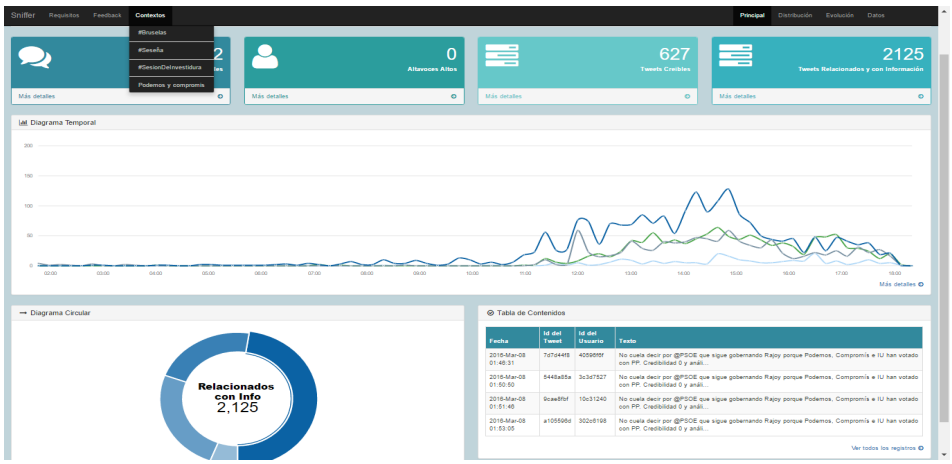

Figura F.6: Vista del contexto

Como puede verse en la figura F.6, se muestra un pequeño resumen en la parte superior que contiene diferentes datos relevantes acerca del contexto. Adicionalmente se muestra también un resumen del resto de vistas asociadas al contexto que se ofrecen al usuario.

Distribución

En esta vista es posible ver la distribución de los tweets asociados a las diferentes clases del sistema. Este diagrama es interactivo, cuando se posiciona el cursor sobre cada una de las diferentes porciones del diagrama se muestra la clase que representa y el total de tweets asociados a esa clase que han sido clasificados durante el experimento ejecutado.

Como parte de todas las vistas asociadas a los contextos, en la parte superior se presenta un menú que permite al usuario volver a la vista inicial o acceder a las diferentes vistas pertenecientes a ese contexto.

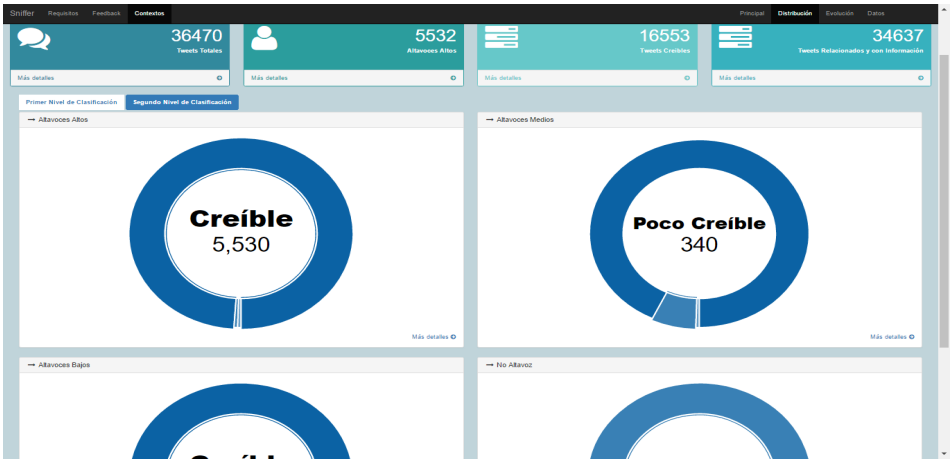

Figura F.7: Vista de distribución

Tablas

En esta vista se pueden apreciar todos los elementos que se han utilizado para la realización del experimento por separado. Al seleccionar cada uno de estas opciones que se presentan se va a crear la tabla con los datos asociados. Hay que remarcar que este proceso de carga de los datos puede demorarse, debido a que se está trabajando con multitud de entradas en la tabla.

Dentro de cada pestaña se muestra una tabla con todos los registros que han sido clasificados en cada nivel correspondiente. De cada uno de estos registros la información que se muestra es la siguiente:

- Fecha de publicación del tweet.
- Identificador asociado al tweet.
- Identificador asociado al usuario.
- Texto transmitido en el tweet.
- Resultado de la primera clasificación (1 es R1, 2 es R2 y así sucesivamente).
- Resultado de la segunda clasificación si procediese (1 es C1, 2 es C2 y 3 es C3)

En las pestañas que representan la división con altavoces se presentan el mismo número de datos cambiándose la clase por el grado de repercusión

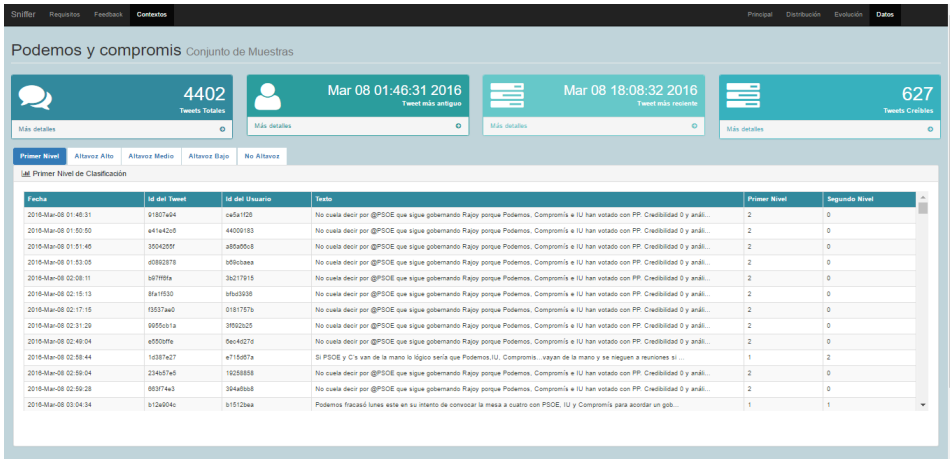

Figura F.8: Vista de tablas

que se le asocia al tweet.

Como puede verse en la imagen F.8, los identificadores tanto de usuario como de tweet han sido transformados de acorde a las necesidades expuestas a lo largo del capítulo 6.

Evolución

En esta vista se puede ver la evolución temporal de los tweets asociados a las diferentes clases del sistema. Este diagrama es interactivo, cuando se posiciona el cursor sobre cada una de las diferentes áreas del diagrama se muestra la clase que representa y el número de tweets publicados dentro de ese intervalo de tiempo y que son los que se representan en ese diagrama.

Esta vista es la que está representada en la figura F.9.

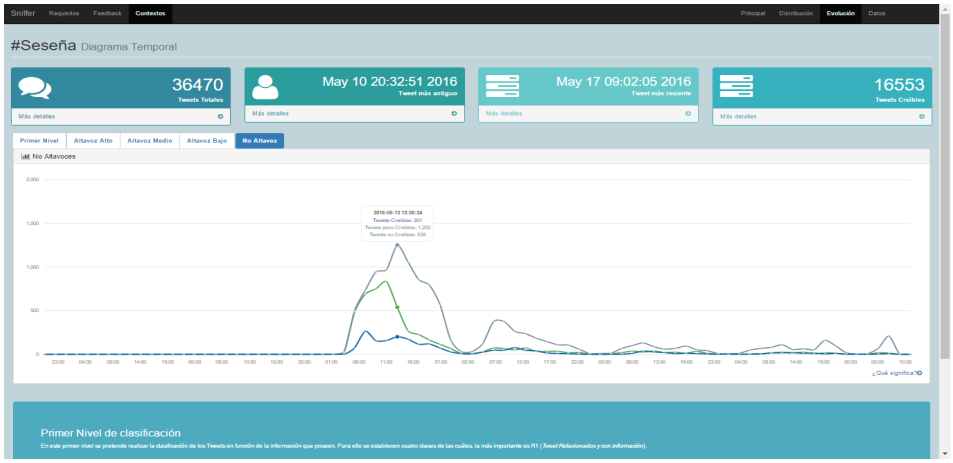

Figura F.9: Vista de evolución

F.2.2 Feedback

Esta vista está destinada a la mejora del conjunto de entrenamiento de los tweets que se utilizan para cada sistema de clasificación. El usuario debe seleccionar entre las opciones que se aportan al desplegar la pestaña. Una vez seleccionada se pulsa en el botón enviar y automáticamente se aporta otro tweet para clasificar.

El objetivo de generar esta vista ha sido dar facilidad al usuario para poder generar de forma rápida y sencilla un conjunto de entrenamiento para su sistema *Sniffer*. Como puede verse, el usuario dispone de información adicional como la clase y el grado de credibilidad que ha asociado la ejecución del programa. Con esto se está permitiendo al usuario *"corregir"* las decisiones del sistema.

Esta perspectiva del sistema es la que se representa a lo largo de la figura F.10.

Figura F.10: Captura de feedback

F.2.3 Requisitos

En esta vista se representan todas las librerías que son necesarias para el funcionamiento del programa. Esta vista está destinada a informar al usuario sobre qué necesita para poder ejecutar el sistema *Sniffer* en su equipo. Esta vista es la que se representa en la figura F.11.

Figura F.11: Vista de los requisitos del sistema